robotron

Anleitung für den
(System-)Programmierer

SCP 1700-Grafikerweiterung SCP-GX

Arbeitsplatzcomputer A 7100 Betriebssystem SCP 1700

Anleitung für den
(System-) Programmierer

Betriebssystem SCP 1700

SCP 1700-Grafikerweiterung
SCP-GX

AC A7100

VEB Robotron-Projekt Dresden

ℂ 1014-0004-2 M 3030

Die vorliegende Systemunterlagendokumentation, Anleitung für den (System-) Programmierer, Betriebssystem SCP 1700, SCP 1700-Grafikerweiterung SCP-GX, entspricht dem Stand von 1/87.

Nachdruck, jegliche Vervielfältigung oder Auszüge daraus sind unzulässig.

Die Ausarbeitung erfolgte durch ein Kollektiv des VEB Robotron-Elektronik Dresden .

Herausgeber:

VEB Robotron-Projekt Dresden, Leningrader Str. 9, Dresden 8010

(C)

Kurzreferat

Das SCP-GX ist die Erweiterung des Betriebssystems SCP 1700 für die Bedienung grafischer E/A-Geräte. Es werden die Bestandteile und Funktionen des SCP-GX vorgestellt und die Nutzung seiner grafischen Möglichkeiten beschrieben.
Weiterhin wird erläutert, wie anwenderspezifische grafische E/A-Geräte in SCP-GX eingebunden werden können.

Ag 706/199/87 – 4243

Inhaltsverzeichnis

		Seite
1.	Übersicht	5
1.1.	Einleitung	5
1.2.	Aufbau der grafischen Systemerweiterung	5
1.3.	Grafik-Modus	6
1.4.	Nutzerprogramme	6
2.	GDOS	7
2.1.	Einleitung	7
2.2.	GDOS-Funktionen	7
2.3.	GDOS-Operationscodes	9
2.4.	Laden der GIOS-Dateien	13
2.5.	Format der Zuweisungstabelle	14
3.	GIOS	16
3.1.	Einleitung	16
3.2.	Aufgabe des GIOS	16
3.3.	Virtuelles Geräteinterface (VGI)	17
3.4.	Bildung einer GIOS-Datei	19
4.	Bedienung des SCP-GX	20
4.1.	Einleitung	20
4.2.	Aktivierung des Grafik-Modus	20
4.3.	Ausschalten des Grafik-Modus	20
4.4.	Initialisierung des Grafik-Modus	21
4.5.	Starten grafischer Programme	22
4.6.	Ermittlung der Speicherplatzanforderungen	22
4.7.	Austestung grafischer Programme unter SCP-GX	23
5.	Fehlermitteilungen	24
6.	Schreiben eines Gerätedrivers	26
Anlage 1	Beispiel fuer einen GIOS - Driver	28
Anlage 2	Virtuelles Geräteinterface (VGI)	36
Anlage 3	Begriffserklärungen	68
Anlage 4	Gerätespezifische Informationen	71
Anlage 5	Drivertestprogramm DRVTST.CMD	101
Anlage 6	Nutzung von SCP-GX unter FORTRAN-77	105
Abkürzungsverzeichnis		108
Sachwortverzeichnis		109

Bildverzeichnis

		Seite
Bild 1:	Grafische Systemerweiterung SCP-GX	5
Bild 2:	SCP-GX Speicherbelegung	21

Tabellenverzeichnis

		Seite
Tabelle 1:	SCP-GX Operationscodes	10
Tabelle 2:	Arbeitsstationsbeschreibungstabelle beim Bildschirmsystem	72
Tabelle 3:	Arbeitsstationsbeschreibungstabelle für Drucker	79
Tabelle 4:	Arbeitsstationsbeschreibungstabelle für Plotter K6418	83
Tabelle 5:	Arbeitsstationsbeschreibungstabelle für Plotter K6411	94

1. Übersicht

1.1. Einleitung

Dieser Abschnitt gibt einen Überblick über den Aufbau der grafischen Systemerweiterung, deren Bestandteile und Funktionen. Nachfolgende Kapitel beschreiben einzelne Teile davon ausführlich.

1.2. Aufbau der grafischen Systemerweiterung

Das SCP-GX ist die grafische Systemerweiterung des SCP 1700-Betriebssystems. SCP-GX besteht aus zwei Teilen, dem hardwareunabhängigen Teil, dem Graphics Device Operating System (GDOS), und dem hardwareabhängigen Teil, dem Graphics Input/Output System (GIOS). Der hardwareabhängige Teil muß für das jeweilige Peripheriegerät implementiert werden. Es ist somit für jedes Gerät ein eigener Driver vorhanden. Sollen grafische Ein-/Ausgaben auf verschiedenen Geräten erfolgen, lädt das SCP-GX den jeweils benötigten Driver. Zur Minimierung des Speicherplatzbedarfs ist immer nur ein Driver jeweils Teil des Betriebssystems. Die Driver für die speziellen grafischen Geräte setzen die Standard-SCP-GX-Rufe in die spezifischen Geräteanweisungen um. Auf diese Weise bietet SCP-GX Geräteunabhängigkeit, da ja die Besonderheiten der grafischen Geräte für das Nutzerprogramm nicht sichtbar sind. Bild 1 stellt die grafische Systemerweiterung dar.

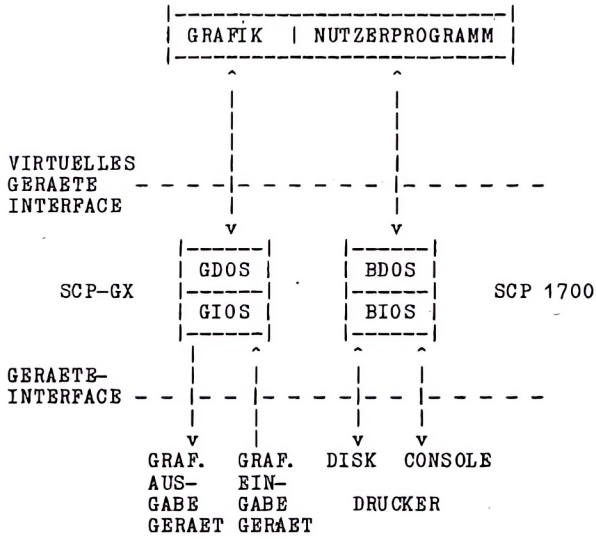

Bild 1: Grafische Systemerweiterung SCP-GX

1.3. Grafik-Modus

Von der Kommandoebene des SCP 1700 aus kann ein spezielles Kommando eingegeben werden, welches das Betriebssystem in den grafischen Modus setzt bzw. rücksetzt (GRAPHICS bzw. GRAPHICS NO). Dieses Kommando bewirkt das Laden von GDOS und des Standardgerätedrivers und die Herstellung geeigneter Verbindungen zum Betriebssystem. Es gestattet dadurch einem Nutzer den Zugriff zu den Grafikgeräten. Wird der Grafik-Modus rückgesetzt, so wird der ganze Speicherplatz freigegeben und der gesamte Speicher steht für Nichtgrafikprogramme zur Verfügung.

1.4. Nutzerprogramme

Man kann die Nutzerprogramme in Assembler oder in einer höheren Programmiersprache, die die SCP-GX-Rufe unterstützt, schreiben. Diese Programme können auf die im SCP übliche Art und Weise assembliert, editiert und geladen werden.

2. GDOS

2.1. Einleitung

Dieser Abschnitt beschreibt das Graphics Device Operating System (GDOS) im Einzelnen, einschließlich der GDOS-Funktionen, der GDOS-Ruffolge und das Laden von Gerätedrivern. Das GDOS enthält den Eintrittspunkt sowohl für die herkömmlichen BDOS-Rufe als auch für die geräteunabhängigen grafischen Funktionen, die durch das Nutzerprogramm gerufen werden können. GDOS stellt die Koordinaten so dar, daß das Programm die Punkte in einem normierten Koordinatenraum angeben kann. Es benutzt gerätespezifische Informationen zum Umsetzen der Normalisierten Koordinaten (NDC) in die entsprechenden Werte für das spezielle Grafikgerät (Gerätekoordinaten DC). Innerhalb eines einzelnen Nutzerprogrammes können vielfältige grafische Geräte unterstützt werden. Die Auswahl des jeweiligen Grafikgerätedrivers erolgt über eine dem Gerät zugeordnete Arbeitsstations-Identifikationsnummer. Beim Eröffnen einer Arbeitsstation wird der entsprechende Gerätedriver in den Speicher geladen. Dabei wird der vorhergehende Driver überlagert. Diese Technik minimiert den Speicherplatzbedarf, da sich zu jeder Zeit nur ein Driver im Speicher befindet. Näheres siehe Abschnitt 2.4. "Laden der GIOS-Dateien".

2.2. GDOS-Funktionen

GDOS führt während der Abarbeitung eines grafischen Nutzerprogrammes 3 Funktionen aus:

- Reagieren auf grafische Anforderungen
- Laden der verlangten Gerätedriver
- Umwandeln der Normalisierten Koordinaten in Gerätekoordinaten

Graph. Anforderungen: Ein Nutzerprogramm erreicht GDOS über den Interrupt 224 mit dem Funktionscode 0473H im Register CX und der Segmentadresse und dem Offset eines Parameterblocks in DS:DX.

Dynamisches Laden: Ein Nutzerprogramm eröffnet eine Arbeitsstation und GDOS bestimmt, ob der geforderte Gerätedriver im Speicher resident ist. Falls nicht, lädt GDOS den Driver von der Diskette und überprüft die grafische Anforderung.

Das Nutzerprogramm übergibt alle grafischen Koordinaten an GDOS als Normalisierte Koordinaten (NDC) in einem Bereich von 0 bis 32767 auf beiden Achsen. Werden vom Gerätedriver Informationen übernommen, wandelt GDOS die NDC-Einheiten in Gerätekoordinaten (DC) um. Der vollständige NDC-Skalenbereich ist immer auf beiden Achsen in vollem Umfang für das grafische Gerät darstellbar. Das sichert, daß alle grafischen Informationen auf der Bildfläche erscheinen, ohne Rücksicht auf die Ausdehnung des Gerätes.
Der Zugriff zu grafischen Funktionen erfolgt über das Virtuelle Geräteinterface (VGI). Das VGI verwendet die herkömmliche BDOS-Ruffolge; das Nutzerprogramm ruft GDOS über Interrupt 224 mit einem Funktionscode 0473H im Register CX. Das Programm übergibt die Argumente an GDOS in einer Parameterliste, die durch die Inhalte von Register DS (Segment) und DX (Offset) adressiert

wird.
Die Parameterliste enthält Adressen auf 5 Parameterbereiche:
- Steuerbereich
- Bereich für die Eingabeparameter
- Bereich für die Koordinaten der Eingabepunkte
- Bereich für die Ausgabeparameter
- Bereich für die Koordinaten der Ausgabepunkte

Die spezifische grafische Funktion, die durch GDOS ausgeführt wird, wird durch einen Operationscode in der Parameterliste angezeigt. GDOS behält das BP-(Basisanzeiger) und DS-(Datensegment) Register bei. Alle anderen Register werden durch das GDOS zerstört.

GDOS-Ruffolge:

- Zugriff über Interrupt 224
- Funktionscode in Register CX = 0473H
- Parameterblockadresse in Register DS (Segment) und DX (Offset)

Parameterblockinhalt:

PB Doppelwortadresse des Steuerbereiches
PB+4 Doppelwortadresse des Eingabeparameterbereiches
PB+8 Doppelwortadresse des Bereiches für Eingabepunkte
PB+12 Doppelwortadresse des Ausgabeparameterbereiches
PB+16 Doppelwortadresse des Bereiches für Ausgabepunkte

Steuerbereich bei Eingabe:

contrl(1) - Operationscode für die Gerätefunktion
contrl(2) - Anzahl der Punkte im Eingabepunktbereich (ptsin)
contrl(4) - Länge des Eingabeparameterbereiches (intin)
contrl(6...n)- Operationscodeabhängig

Eingabeparameterbereich

intin Bereich der Eingabeparameter
(Die Länge des Bereiches ist operationscodeabhängig und in contrl(4) spezifiziert)

Eingabepunktbereich:

ptsin Bereich der Eingabepunkte
(Jeder Punkt ist durch ein X- und Y-Koordinatenpaar bestimmt, gegeben in Normalisierten Koordinaten zwischen 0 und 32767; Anzahl der Koordinaten = contrl(2)*2)

Steuerbereich bei Ausgabe:

contrl(3) - Anzahl der Punkte im Ausgabepunktbereich (ptsout)

contrl(5) - Zahl der Elemente im Ausgabeparameterbereich (intout)

contrl(6...n)- operationscodeabhängig

Ausgabeparameterbereich

intout Bereich der Ausgabeparameter
(Die Länge des Bereiches ist operationscodeabhängig)

Ausgabepunktbereich:

ptsout Bereich der Ausgabepunkte
(Jeder Punkt ist durch ein X- und Y-Koordinatenpaar, gegeben in Normalisierten Koordinaten zwischen 0 und 32767, spezifiziert.) Der Bereich muß größer sein als der größtmögliche Wert von contrl(3)*2.

Anmerkung:

Alle Bereichselemente sind vom Typ Integer (2 Bytes). Alle Bereiche beginnen bei 1, d.h. die Doppelwortadresse von PB zeigt auf das erste Element des Steuerbereiches (contrl(1)). Die Bedeutung der Ein- und Ausgabeparameterbereiche ist abhängig vom Operationscode. Siehe im Anhang 2, "Virtuelles Geräteinterface" nach Einzelheiten.

2.3. GDOS-Operationscodes

Tabelle 1 faßt die GDOS-Operationscodes zusammen. Siehe im Anhang 2 nach der detaillierten Beschreibung der Operationscodes, einschließlich der Parameter.

Tabelle 1: SCP-GX Operationscodes

Operationscode	Beschreibung
1	OPEN WORKSTATION Initialisiert ein grafisches Gerät. (Lädt den Driver, falls notwendig.) Liefert Arbeitsstationsbeschreibungstabelle.
2	CLOSE WORKSTATION Stoppt die grafische Ausgabe zu einer Arbeitsstation.
3	CLEAR WORKSTATION Löschen des Bildschirmgerätes, Seitenvorschub bei Druckern und neues Blatt bei Plottern.
4	UPDATE WORKSTATION Sofortige Ausführung aller im Befehlspuffer gespeicherten Grafikanforderungen.
5	ESCAPE Erlauben spezieller geräteabhängiger Operationen. ESCAPE-Funktionen: (Die Funktions-ID wird in der Parameterliste dargestellt.) ID 1 INQUIRE ADDRESSABLE CHARACTER CELLS Abfragen der adressierbaren Zeilen und Spalten. 2 ENTER GRAPHICS-MODUS Einstellen des Grafik-Modus. 3 EXIT GRAPHICS MODUS Aufheben des grafischen Modus. 4 CURSOR UP Bewegen des Cursors um eine Zeile nach oben. 5 CURSOR DOWN Bewegen des Cursors um eine Zeile nach unten. 6 CURSOR RIGHT Bewegen des Cursors um eine Spalte nach rechts. 7 CURSOR LEFT Bewegen des Cursors um eine Spalte nach links. 8 HOME CURSOR Bewegen des Cursors zur Ausgangsstellung. (Zeile 1, Spalte 1). 9 ERASE TO END OF SCREEN Löschen von der augenblicklichen Cursorposition bis zum Bildschirmende. 10 ERASE TO END OF LINE Löschen von der augenblicklichen Cursorposition bis zum Zeilenende. 11 DIRECT CURSOR ADDRESS Bewegen des Cursors zur spezifizierten Zeile und Spalte.

Tabelle 1: Fortsetzung

Operationscode	Beschreibung
	ID 12 OUTPUT CURSOR ADDRESSABLE TEXT Textausgabe ab augenblicklicher Cursorposition. 13 REVERSE VIDEO ON Anzeigen des nachfolgenden Textes im Invers-Modus 14 REVERSE VIDEO OFF Anzeigen des nachfolgenden Textes im Normalmodus. 15 INQUIRE CURRENT CURSOR ADDRESS Abfragen der aktuellen Cursoradresse. 16 INQUIRE TABLET STATUS Statusabfrage für das Grafik-Tablett 17 HARDCOPY Hardcopy veranlassen. 18 PLACE CURSOR AT LOCATION Bewegen des Cursors zur angegebenen Position. 19 REMOVE CURSOR Der Cursor wird nicht angezeigt. 20-100 gerätespezifisch
6	POLYLINE Ausgabe Polyline (Linienzug zwischen einer Folge von Koordinatenpunkten)
7	POLYMARKER Ausgabe von Polymarker (Markersymbole an gegebenen Koordinatenpunkten darstellen)
8	TEXT Ausgabe von Text, beginnend bei der angegebenen Position.
9	FILLED AREA Ausgabe und Füllen eines Polygons.
10	CELL AREA Ausgabe einer Zellmatrix.
11	GENERALIZED DRAWING PRIMITIVE Ausgabe eines Verallgemeinerten Darstellungselementes. ID 1 Balken 2 Kreisbogen 3 Kreissektor 4 Kreis 5 Grafische Zeichen 6-7 Reserviert für spätere Erweiterungen 8-10 Ungenutzt und verfügbar

Tabelle 1: Fortsetzung

Oerationscode	Beschreibung
12	SET CHARACTER HEIGHT Festlegen der Höhe von Grafikzeichen.
13	SET CHARACTER UP VECTOR Festlegen des Vektors der Schriftrichtung.
14	SET COLOR REPRESENTATION Definieren der Farbe, verbunden mit einem Farbindex.
15	SET POLYLINE LINETYPE Festlegen der Linienart für Polyline.
16	SET POLYLINE LINEWIDTH Festlegen der Linienstärke.
17	SET POLYLINE COLOR INDEX Festlegen der Farbe für Polyline.
18	SET POLYMARKER TYPE Festlegen des Markertypes für Polymarker.
19	SET POLYMARKER SCALE Festlegen der Größe für Polymarker.
20	SET POLYMARKER COLOR INDEX Festlegen der Farbe für Polymarker.
21	SET TEXT FONT Festlegen der geräteabhängigen Textart.
22	SET TEXT COLOR INDEX Festlegen des Farbindexes des Textes.
23	SET FILL INTERIOR STYLE Festlegen der Füllart für Füllgebiete. (z. B. leer, voll, gemustert, schraffiert)
24	SET FILL STYLE INDEX Festlegen des Füllindex für Füllgebiete.
25	SET FILL COLOR INDEX Festlegen der Farbe für Füllgebiete.
26	INQUIRE COLOR REPRESENTATION Zurücksenden der Farbdarstellung.
27	INQUIRE CELL ARRAY Zurücksenden der Definition der Zellmatrix.

Tabelle 1: Fortsetzung

Operationscode	Beschreibung
28	INPUT LOCATOR Zurücksenden des Locatorwertes.
29	INPUT VALUATOR Zurücksenden des Valuatorwertes.
30	INPUT CHOICE Zurücksenden des Wertes des Choicegerätes.
31	INPUT STRING Zurücksenden einer Zeichenfolge.
32	SET WRITING MODE Festlegen des aktuellen Schreibmodus. (Ersetzen, Überschreiben, Komplement, Löschen)
33	SET INPUT MODE Festlegen des Eingabemodus. (Request oder Sample)
34	INPUT STROKE Eingabe einer Folge von Locatorwerten

2.4. Laden der GIOS-Dateien

Das "Virtuelle Geräteinterface" bezieht sich auf die grafischen Geräte als Arbeitsstationen. Bevor ein grafisches Gerät benutzt werden kann, muß es zuerst mit der OPEN WORKSTATION - Operation initialisiert werden. Diese Operation initialisiert das Gerät mit ausgewählten Eigenschaften, wie z.B. Linientyp, Farbe usw., und sie sendet auch die Informationen (Arbeitsstationsbeschreibungstabelle) über das Gerät an GDOS zurück.
Wenn die OPEN WORKSTATION - Operation angewiesen ist, bestimmt GDOS, ob die entsprechende GIOS-Datei, der Gerätedriver, schon im Speicher ist oder nicht. Dies geschieht durch Vergleich der Arbeitsstations-ID, die im OPEN WORKSTATION - Ruf angegeben ist, mit der WORKSTATION-ID des Gerätes, dessen Driver augenblicklich geladen ist.
Wenn es eine Übereinstimmung gibt (die entsprechende GIOS-Datei ist im Speicher) wird das OPEN WORKSTATION - Gesuch sofort bearbeitet. Ist dies nicht der Fall, muß die benötigte GIOS-Datei geladen werden. Um sie zu finden, nimmt GDOS auf eine Datenstruktur Bezug, Zuweisungstabelle genannt, die Informationen über die verfügbaren Gerätedriver und deren Dateinamen enthält.
GDOS sucht in der Zuweisungstabelle den ersten Zuweisungssatz mit einer Gerätenummer, die der Arbeitsstations-ID, nach der in dem OPEN WORKSTATION - Ruf gefragt wird, entspricht. Falls die gesuchte Gerätenummer gefunden ist, lädt GDOS die neue GIOS-Datei dorthin, wo die vorhergehende geladen war. Wenn das Laden abgeschlossen ist, beendet GDOS die OPEN WORKSTATION - Operation und kehrt zum rufenden Programm zurück. Wird ein neuer Driver

benötigt, und in der Zuweisungstabelle nichts gefunden, dann kehrt GDOS zurück ohne einen Driver zu laden, und das vorhergehende grafische Gerät setzt seine Arbeit als eröffnete Arbeitsstation fort.

2.5. Format der Zuweisungstabelle

Die Zuweisungstabelle besteht vollständig aus Text und kann mit irgendeinem Texteditor erstellt oder verändert werden. Sie muß in einer Datei mit dem Namen ASSIGN.SYS auf dem Laufwerk zu finden sein, welches im Kommando GRAPHICS angegeben wurde, oder auf dem aktuellen Laufwerk, falls kein Laufwerk im GRAPHICS-Kommando angegeben wurde. Für jeden Gerätedriver gibt es einen Zuweisungssatz, der die Gerätenummer enthält, die der Arbeitsstations-ID des Gerätes entspricht, und den Namen der Datei, die den dazugehörigen Grafik-Gerätedriver enthält. Der Name der Gerätedriverdatei kann irgendein zulässiger SCP 1700 - Dateiname sein.

Das Format für die Eintrittspunkte in der Zuweisungstabelle ist:

```
DDXd:Dateiname;Kommentar

DD = logische Gerätenummer
X  = Leerzeichen
d  = Laufwerkscode

Dateiname = Driver-Dateiname
            (gültiger bis zu acht Zeichen langer Dateiname und
            Dateityp, .SYS wird als Standard angenommen)

Kommentar = irgendeine Zeichenfolge
```

Gültige Eintrittspunkte in der Tabelle sind zum Beispiel:

```
 11 A:PLOTTER;Plotter
  1 B:MONITC;System Console
 21 A:PRINT13;Printer
  2 C:DRIVER.DRV
 12 DRIVER2.SYS
```

Anmerkungen:

- Der Driver-Dateiname kann jeden beliebigen Dateityp haben, es wird jedoch .SYS vorausgesetzt, wenn das Dateityp-Feld leer ist. Das Laufwerk, das im GRAPHICS-Befehl spezifiziert ist, wird als das aktuelle für die Driver-Dateinamen benutzt, wenn im Eintrittspunkt nicht explizit auf ein Gerät Bezug genommen wird. Zusätzliche Leerzeichen können eingefügt werden.
- Die folgenden Vereinbarungen für die zugewiesenen Gerätedriver-Nummern (Arbeitsstations-ID's) für grafische Geräte sichern einen maximalen Grad an Geräteunabhängigkeit innerhalb der Nutzerprogramme. Die Vereinbarung für Driver-Nummern ist:
  1-10 CRT
  11-20 Plotter
  21-30 Drucker
  31-40 andere Geräte
- Wenn der Grafik-Modus durch den GRAPHICS-Befehl erlaubt ist, weist SCP-GX Speicher für den ersten Gerätedriver in der Zuweisungstabelle zu. Dieser Driver wird als Standard-Gerätedriver

angenommen. Folglich veranlaßt GDOS, daß alle neuen Driver in denselben Bereich geladen werden, wo der Speicher für den ursprünglichen Gerätedriver zugewiesen war. Es ist zu sichern, daß der erste Driver in der Zuweisungstabelle der größte zu ladende Driver ist, so daß genügend Speicherplatz durch den SCP-GX-Lader für alle folgenden Driver zugewiesen wird. SCP-GX gibt einen Fehler an das rufende Programm (und der neue Driver ist nicht geladen), falls der Versuch gemacht wird, einen Driver zu laden, der größer als der aktuelle Driver ist.

3. GIOS

3.1. Einleitung

Das Graphic Input/Output System (GIOS) ist ähnlich dem Basic-I/O-System (BIOS). Es enthält den gerätespezifischen Code, der die Schnittstelle zwischen einem speziellen Grafikgerät und dem GDOS realisiert.
SCP-GX verlangt für jedes grafische Gerät einen eigenen Gerätedriver. Die einzelnen Driver, die in den Speicher geladen werden, wenn sie durch das Nutzerprogramm angefordert werden, werden GIOS-Dateien genannt. Obwohl ein einzelnes Programm mehrere grafische Geräte benutzen kann, lädt GDOS immer nur eine GIOS-Datei zu einem Zeitpunkt.
GIOS führt die grafischen Funktionen des SCP-GX, entsprechend der Hardware-Möglichkeiten des grafischen Gerätes aus. In einigen Fällen emuliert ein Gerätedriver GDOS-Funktionen, welche durch die Gerätehardware nicht direkt ausgeführt werden können. Im folgenden werden Informationen geliefert, die helfen sollen, eigene Driver zu schreiben. Das Virtuelle Geräteinterface (VGI) im Anhang 2 definiert dazu alle erforderlichen Funktionen und Parameter. Anhang 1 enthält ein Beispiel eines Gerätedrivers.

3.2. Aufgabe des GIOS

Wie schon erwähnt, besteht SCP-GX aus zwei Komponenten: dem Graphics Device Operating System (GDOS) und dem Graphics Input/Output System (GIOS). GDOS enthält die geräteunabhängigen grafischen Funktionen, während GIOS geräteabhängigen Code enthält. Dies erlaubt, Anwendungen unabhängig von den spezifischen Geräten, die mit dem System verbunden sind, zu entwickeln. GIOS ist analog dem BIOS, betrifft aber nur die grafischen Geräte. GIOS enthält für jedes grafische Gerät im System eine Datei, den Gerätedriver.
Ein Unterschied zwischen GIOS und BIOS ist, daß alle Gerätedriver, die im BIOS enthalten sind, gleichzeitig im Speicher residient sind, wohingegen beim GIOS nur ein grafischer Gerätedriver zu einem Zeitpunkt residnt ist. Das heißt zu einem Zeitpunkt ist nur ein grafisches Gerät aktiv. Das aktive Gerät kann jedoch durch eine Forderung vom Nutzerprogramm aus gewechselt werden. GDOS sichert, daß der richtige Driver im Speicher ist, wenn er benötigt wird.
Jede GIOS-Datei nutzt die grafischen Fähigkeiten der Geräte, um die grafischen Funktionen des GDOS auszuführen. In einigen Fällen, in denen Grafik-Befehle für SCP-GX durch den Gerätedriver nicht direkt ausführbar sind, muß der Driver die fehlende Geräteintelligenz in bestimmtem Umfang ausgleichen. Zum Beispiel, falls ein Plotter keine gestrichelte Linie erzeugen kann, muß der Driver diese durch eine Folge von Polylines ersetzen, und diese dem Plotter übergeben.

3.3. Virtuelles Geräteinterface (VGI)

Die Gerätedriver müssen das Virtuelle Geräteinterface (VGI) des SCP-GX realisieren.
Das VGI spezifiziert sowohl die Ruffolge zum Zugriff auf die Gerätedriverfunktionen als auch die Syntax und Semantik der Datenstrukturen, die mit dem Interface in Verbindung stehen. Das Nutzerprogramm übergibt die Argumente an die Gerätedriver in einer Parameterliste, die durch die Inhalte der Register DS (Segment) und DX (Offset) angezeigt wird.
Die Parameterliste besteht aus 5 Bereichen:
- Steuerbereich
- Bereich für die Eingabeparameter
- Bereich für die Koordinaten der Eingabepunkte
- Bereich für die Ausgabeparameter
- Bereich für die Koordinaten der Ausgabepunkte

Das Nutzerprogramm spezifiziert die grafische Funktion, die von einem Gerätedriver ausgeführt wird, mit einem Operationscode im Steuerbereich. Erinnert sei daran, daß GDOS die BP- und DS-Register rettet. Alle anderen Register können verändert sein, wenn von GDOS zurückgekehrt wird.

Die Ruffolge für die Gerätedriver wird wie folgt zusammengefaßt.
- Zugriff über Interrupt 224
- Funktionscode (im Reg. CX=0473H)
- Parameterblockadresse im Register DS (Segment) und DX (Offset)

Parameterblockinhalt:

PB	Doppelwortadresse des Steuerbereiches
PB+ 4	Doppelwortadresse des Eingabeparameterbereiches
PB+ 8	Doppelwortadresse des Bereiches für Eingabepunkte
PB+12	Doppelwortadresse des Ausgabeparameterbereiches
PB+16	Doppelwortadresse des Bereiches für Ausgabepunkte

Steuerbereich bei Eingabe:

contrl(1)	-Operationscode für die Driverfunktion
contrl(2)	-Anzahl der Punkte im Eingabepunktbereich (ptsin)
contrl(4)	-Länge des Eingabeparameterbereiches (intin)
contrl(6-n)	-Operationscodeabhängig

Eingabeparameterbereich:

intin Bereich der Eingabeparameter (die Länge des Bereiches ist operationscodeabhängig und in contrl(4) spezifiziert)

Eingabepunktbereich:

ptsin Bereich der Eingabepunkte (jeder Punkt wird durch ein X- und Y-Koordinatenpaar spezifiziert, gegeben in Normalisierten Koordinaten zwischen 0 und 32767, Anzahl der Koordinaten contrl(2)*2)

Steuerbereich bei Ausgabe:

contrl(3)	-Anzahl der Punkte im Ausgabepunktbereich (ptsout)
contrl(5)	-Anzahl der Elemente im Ausgabeparameterbereich

(intout)
contrl(6-n) -Operationscodeabhängig

Ausgabeparameterbereich:

intout Bereich der Ausgabeparameter
(Länge des Bereiches ist operationscodeabhängig)

Ausgabepunktbereich:

ptsout Bereich der Ausgabepunkte
(jeder Punkt wird durch ein X- und Y-Koordinatenpaar spezifiziert, gegeben in Normalisierten Koordinaten zwischen 0 und 32767; Länge größer als der größtmöglichste Wert von contrl(5)*2)

Anmerkungen:

Alle Bereichselemente sind vom Typ Integer (2 Bytes). Alle Bereiche beginnen bei 1, d.h. die Doppelwortadresse von PB zeigt auf das erste Element des Steuerbereiches (contrl(1)). Die Bedeutung der Ein- und Ausgabeparameterbereiche ist vom Operationscode abhängig. Siehe im Anhang 2, "Virtuelles Geräteinterface", nach Einzelheiten.

Das Nutzerprogramm übergibt alle grafischen Koordinaten als NDC-Koordinaten ans GDOS und das GDOS übergibt sie als Gerätekoordinaten (DC) an den Driver.

Die notwendigen Informationen, um die vom Nutzer gesendeten NDC-Koordinaten in DC-Koordinaten zu transformieren, erhält das GDOS beim Eröffnen der Arbeitsstation als Rückinformation vom Driver.

Der gesamte NDC-Raum wird in beiden Achsrichtungen auf den DC-Raum abgebildet. Dies sichert, daß alle grafischen Informationen auf der Anzeigefläche ohne Berücksichtigung der aktuellen Gerätedimension sichtbar sind.

Falls das Gerät ein Anzeigeverhältnis hat, das nicht 1:1 ist (d.h. die Anzeigefläche ist nicht quadratisch), und Verzerrungen verhindert werden sollen, dann müssen in den Nutzerprogrammen für die X- und Y-Richtung unterschiedliche Maßstabsfaktoren genutzt werden, um die Asymmetrie des Gerätes auszugleichen. Wird zum Beispiel ein CRT-Gerät mit einem Anzeigeverhältnis von 3:4 (vertikal:horizontal) benutzt, dann muß ein Bild mit 4000 NDC-Einheiten vertikal und 3000 NDC-Einheiten horizontal gezeichnet werden, um ein Quadrat auf dem Bildschirm zu erzeugen. Das bedeutet, der Maßstabsfaktor für die vertikale Dimension ist 4/3 zur horizontalen Richtung.

Details über die virtuelle Geräteschnittstelle, einschließlich der geforderten und optionalen Funktionen und Argumente sind im Anhang 2 "Virtuelles Geräteinterface" enthalten.

3.4. Bildung einer GIOS-Datei

Gerätedriverdateien, die ein Teil des GIOS sind, müssen im .CMD-Standardformat sein, so daß sie durch GDOS geladen werden können. Diese Dateien können zu .SYS-Dateien (dem Standard-Dateityp für SCP-GX GIOS-Dateien) umbenannt werden. Gerätedriver können in einer beliebigen Sprache geschrieben werden, sofern die Funktionen und Parameterübergabekonventionen mit den genannten VGI-Spezifikationen vereinbar sind.
GIOS-Dateien können bis zu 8 Zeichen im Dateinamen enthalten und .SYS für den Dateityp. Außerdem muß der Driver in die Zuweisungstabelle eingeordnet werden. Dies ist eine Textdatei mit dem Namen ASSIGN.SYS auf dem aktuellen oder speziell zugewiesenen Laufwerk.
Jeder Geräteeintrittspunkt in der Zuweisungstabelle hat die Form:

DDXd:Dateiname;Kommentar mit

DD = logische Gerätenummer
X = Leerzeichen
d = Laufwerkscode
Dateiname = Driver-Dateiname (gültiger SCP 1700 Dateiname bis zu 8 Zeichen lang und der Dateityp, .SYS-Erweiterung ist Standard)
Kommentar = irgendeine Zeichenfolge

Siehe Kapitel 2.5. Format der Zuweisungstabelle.

4. Bedienung des SCP-GX

4.1. Einleitung

In diesem Abschnitt wird erläutert, wie die Grafikerweiterung zu SCP 1700 aktiviert bzw. deaktiviert werden kann.

4.2. Aktivierung des Grafik-Modus

Vor dem ersten Start eines Grafikprogrammes muß man SCP-GX durch Eingabe des GRAPHICS-Kommandos aktivieren. Dadurch werden dem Nutzerprogramm die Grafikfunktionen verfügbar gemacht.
Vor der Eingabe des GRAPHICS-Kommandos müssen gerätespezifische Besonderheiten beachtet werden (z. B. Laden der Firmware für das grafische Bildschirmsystem).
Die Benutzung des GRAPHICS-Kommandos, als Antwort auf die Systemmeldung, die beim Start des Systems erscheint, geschieht wie folgt:
A>GRAPHICS <drive>:
Gibt man
A>GRAPHICS A:
ein, so sucht SCP-GX in der ASSIGN.SYS-Datei auf dem Laufwerk A nach der Zuweisungstabelle, die die verfügbaren Driver enthält. (Beide, die ASSIGN.SYS-Datei und die Gerätedriver müssen auf dem spezifizierten Laufwerk sein.)
Gibt man
A>GRAPHICS B:
ein, sucht SCP-GX auf dem Laufwerk B.
Wird kein Gerät im GRAPHICS-Kommando angegeben, wird auf dem aktuellen Laufwerk nach der ASSIGN.SYS- Datei gesucht.
Wenn man SCP-GX korrekt installiert hat, erhält man folgende Mitteilung:
SCP-GX installed; name is dddd bytes long at xxxx:oooo
Dies bedeutet, SCP-GX und der Standard-Gerätedriver (name) wurden geladen. "dddd" ist die exakte Größe des Gerätedrivers name, der ab xxxx:oooo im Hauptspeicher steht.

4.3. Ausschalten des Grafik-Modus

Werden keine grafischen Funktionen mehr benötigt, kann der grafische Modus durch Eingabe von
A>GRAPHICS NO
ausgeschalten werden.
Dieses Kommando gibt den gesamten Speicherbereich, der für SCP-GX reserviert war, für die Nutzerprogramme frei. Danach erhält man folgende Mitteilung:
SCP-GX not installed
Diese Mitteilung kann man auch erhalten, wenn etwas das Installieren von SCP-GX verhindert.

4.4. Initialisierung des Grafik-Modus

Beim Eintritt in den Grafik-Modus führt das Betriebssystem verschiedene Aktionen durch.
Zuerst wird das GDOS zusammen mit dem Standardgerätedriver (erster Gerätedriver in der Zuordnungstabelle) in den Speicher gebracht.
Als nächstes wird ein spezieller grafischer Interruptvektor zur Interruptnummer 224, Adresse 0380H, gebracht, der zum Eintrittspunkt des GDOS zeigt. Dies bewirkt, daß der Softwareinterrupt INT 224 die Steuerung dem GDOS übergibt. GDOS-Rufe werden bearbeitet, während BDOS-Rufe an das BDOS weitergeleitet werden.
Schließlich wird die Steuerung zum System zurückgegeben, welches auf die nächste Bedienereingabe wartet. Bemerkt sei, daß ein Systemreset (CTRL/C) die Grafik-Modus-Initialisierung nicht zerstört. Ein Hardware-Booten würde die Grafik-Modus-Initialisierung aufheben, und das GRAPHICS-Kommando muß erneut eingegeben werden. Bild 2 zeigt die Anordnung der Bestandteile des SCP-GX nach Grafik-Modus-Initialisierung.
Nach der Ausführung des GRAPHICS NO - Kommandos ist der Speicher, der vom GDOS und vom GIOS belegt wurde, für die Nutzerprogramme freigegeben. Der Interruptvektor 224 ist wieder im Originalzustand und das System erwartet eine Bedienereingabe.

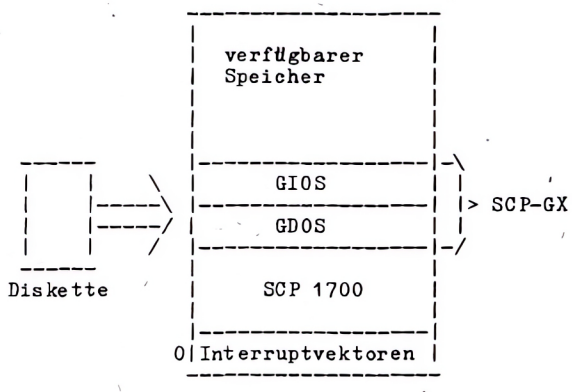

Bild 2: SCP-GX Speicherbelegung

4.5. Starten grafischer Programme

Um die grafischen Eigenschaften des SCP-GX nutzen zu können, müssen folgende Bedingungen erfüllt sein:

1. Die Grafikrufe im Nutzerprogramm müssen den SCP-GX-Aufrufbedingungen genügen. Dies beinhaltet einen Ruf an das Betriebssystem über Interrupt 224 mit dem Funktionscode 0473H im Register CX. Außerdem muß bei Ausführung des Rufes im Registerpaar DS:DX eine Doppelwortadresse bereitgestellt werden, die auf einen Parameterblock verweist. Diese Liste enthält die Eingangsinformationen für einen SCP-GX-Ruf und freie Speicherpositionen zur Übernahme der Rückgabeinformationen von SCP-GX an das Nutzerprogramm. Details dazu sind in den Abschnitten GDOS und GIOS und im Anhang 2 "Virtuelles Geräteinterface" enthalten. Beim Aufruf des SCP-GX werden nur die Register BP und DS gerettet; das Richtungsflag kann verändert sein.
2. Man muß sichern, daß genügend Stackbereich für die SCP-GX-Operationen verfügbar ist. Der Algorithmus zum Bestimmen des geforderten Stack-Speicherplatzes wird unten beschrieben.
3. Man muß sichern, daß die geforderten Gerätedriver auf dem Laufwerk, das im GRAPHICS-Kommando angegeben wurde (oder dem aktuellen Standardlaufwerk, falls im GRAPHICS-Kommando kein Gerät angegeben ist), vorhanden sind, wenn das Programm abgearbeitet wird. Außerdem muß die Zuweisungstabelle die Namen der Gerätedriver enthalten, sowie eine dem gewünschten Gerätedriver entsprechende logische Gerätenummer bzw. Arbeitsstation-ID.
4. Nach dem erfolgreichen Übersetzen und Verbinden des Nutzerprogrammes, kann man es wie jedes andere Programm starten, vorausgesetzt, daß SCP-GX aktiv ist. Die Grafik muß durch das Kommando
   A>GRAPHICS<drive>:
   aktiviert werden.

4.6. Ermittlung der Speicherplatzanforderungen

Für die Ermittlung der für den Start eines Nutzerprogrammes benötigte Stackgröße gilt folgende Berechnung.

SCP-GX Stackbedarf:
OPEN WORKSTATION-Ruf = annähernd 400 Bytes
alle anderen Rufe = Länge von ptsin + 64

Ptsin ist der Bereich der Eingabekoordinaten, der vom Nutzerprogramm an den Gerätedriver übergeben wird. (2 Worte pro Punkt)
Der Stackbedarf ist das Maximum der beiden Werte. Dieser Stackspeicherplatz muß im Stackbereich des Nutzerprogrammes verfügbar sein.
Der Speicherplatz, den das GDOS benötigt, ist kleiner als 3K Bytes. Er wird zugewiesen, wenn das GRAPHICS-Kommando ausgeführt wird. Zur gleichen Zeit wird auch der Speicherplatz für den Standardgerätedriver zugewiesen. Der Standardgerätedriver muß der mit dem größten Speicherplatzbedarf sein, so daß andere Driver, die während der Abarbeitung des Nutzerprogrammes geladen werden, ausreichend Speicherplatz im GDOS-Bereich zur Verfügung haben.

4.7. Austestung grafischer Programme unter SCP-GX

Grafische Programme können wie jedes andere Nutzerprogramm unter Verwendung von DDT86 oder anderer Testhilfen ausgetestet werden. Nachdem das GRAPHICS-Kommando ausgeführt wurde, werden die Standardgerätedriver und das GDOS direkt oberhalb des SCP 1700 geladen. Das grafische Nutzerprogramm wird wie üblich ans Ende des Nutzerbereichs geladen.

5. Fehlermitteilungen

Im allgemeinen werden Register und Flags (einschließlich Richtungsflag) bei Rückkehr von einem Ruf an das SCP-GX nicht wieder hergestellt. Das GIOS rettet die Register DS, SS, CS, BP und SP. Das SCP-GX verändert keine vom GIOS übergebenen Register außer während des Kommandos OPEN WORKSTATION. In diesem Fall wird das Register AX zur Übergabe von Statusinformationen modifiziert. (Die Flags werden durch dieses Kommando ebenfalls modifiziert.) Der Inhalt von AX hat bei der Rückkehr vom Ruf OPEN WORKSTATION folgende Bedeutung:

AL=0 Das Öffnen der Arbeitsstation war erfolgreich.

AL=255 Fehler - Der Gerätedriver ist nicht geladen. In diesem Fall hat AH eine folgende Bedeutung:

AH
- 0 ASSIGN.SYS nicht gefunden
- 1 Syntaxfehler in ASSIGN.SYS
- 2 Geräte-ID in ASSIGN.SYS nicht gefunden
- 3 Close-Fehler für ASSIGN.SYS
- 4 Gerätedriverdatei, spezifiziert in ASSIGN.SYS, wird nicht gefunden.
- 5 Gerätedriverdatei, spezifiziert in ASSIGN.SYS, ist leer.
- 6 Syntaxfehler in einer Datei, die in ASSIGN.SYS spezifiziert ist (z.B. absolutes Codesegment oder kein .CMD-Format)
- 7 Nicht genügend Platz für die spezifizierte Datei

Tritt während der Übertragung einer GIOS-Datei beim OPEN WORKSTATION-Ruf ein Lesefehler auf, wird das Nutzerprogramm beendet, eine Mitteilung wird ausgeschrieben und die Steuerung an den Monitor zurückgegeben.
Folgende Fehlerausschriften können als Antwort auf die SCP-GX-Rufe erscheinen:

- SCP-GX CS:IP GIOS load error on Id xxxxh (hex)
  Während der Übertragung des Gerätedrivers von der Platte trat ein Fehler auf. Die Werte von CS:IP und die Geräte-ID werden ebenfalls angezeigt.

- SCP-GX CS:IP GIOS invalid
  Der zur Zeit geladene Gerätedriver ist ungültig; tritt eventuell nach einem Ladefehler auf, wenn das Nutzerprogramm das Kommando OPEN WORKSTATION nicht als erste grafische Operation ausführt.

- SCP-GX CS:IP Illegal function: (CX)
  In CX wurde ein ungültiger Funktionscode (#0473H) angegeben. Der fehlerhafte Code wird ausgeschrieben.

Die folgenden Fehler können während der Ausführung des GRAPHICS-Kommandos auftreten.

- d:ASSIGN .SYS not found
  ASSIGN.SYS ist auf dem Gerät d: nicht vorhanden.

- d:ASSIGN .SYS syntax error
  Die erste Zeile in ASSIGN.SYS ist nicht erkennbar.

- d:ASSIGN .SYS close error
  Während des Schließens von ASSIGN.SYS trat ein Fehler auf.

- d:ffffffff.xxx not found
  Die GIOS-Datei ffffffff.xxx, die in ASSIGN.SYS spezifiziert ist, wurde auf dem Gerät d: nicht gefunden.

- d:ffffffff.xxx empty
  Die GIOS-Datei ffffffff.xxx wurde gefunden, enthält aber keine Daten.

- d:ffffffff.xxx contains absolute segment
  Die GIOS-Datei ffffffff.xxx enthält ein absolutes Segment, was unter SCP-GX nicht erlaubt ist.

- d:ffffffff.xxx close error
  Während des Schließens der GIOS-Datei ffffffff.xxx trat ein Fehler auf.

- d:ffffffff.xxx load error
  Während des Lesens der GIOS-Datei ffffffff.xxx trat ein Fehler auf.

- Not enough memory for SCP-GX
  Es ist nicht genügend Speicherplatz für SCP-GX und die Standard-GIOS-Datei vorhanden.

6. Schreiben eines Gerätedrivers

SCP-GX wird zusammen mit Drivern für die in Anhang 4 genannten Grafikgeräte ausgeliefert. Enthält eine spezielle Nutzerkonfiguration keine anderen Grafikgeräte, muß nur die Zuweisungstabelle ASSIGN.SYS entsprechend der gewünschten Gerätenummernzuordnung eingerichtet werden.
Wird jedoch ein Gerät nicht unterstützt, muß man dafür ein Driverprogramm schreiben, das den VGI-Spezifikationen entspricht.
Der Driver muß die als gefordert gekennzeichneten Funktionen der VGI-Spezifikation enthalten, und muß den VGI-Parameterübergabebedingungen genügen. In einigen Fällen ist die mittels VGI spezifizierte Fähigkeit in dem grafischen Gerät nicht vorhanden und muß durch das Driverprogramm nachempfunden werden. Zum Beispiel können gestrichelte Linien durch den Driver generiert werden, falls sie nicht direkt vom Gerät erzeugt werden können. Die vollständige VGI-Spezifikation wird im Anhang 2 beschrieben. Die Parameter-Übergabebedingungen werden im Abschnitt 2 "GDOS" und im Abschnitt 3 "GIOS" behandelt.
Gerätedriver werden vom SCP-GX mittels "CALLF" aufgerufen und sollten mit einem "RETF" beendet werden. Der Driver muß für seinen internen Gebrauch seinen eigenen Stack benutzen, abgesehen von einem kleinen Vorspann zum Retten der Zustände des rufenden Programmes

```
CGROUP          GROUP           ;DRIVER_CODE
DRIVER_CODE     CSEG
                PUBLIC DRIVER
DRIVER:         MOV     AX,SP           ;RETTEN DER STACK-ZEIGER
                MOV     BX,SS
;BEMERKT SEI, DASS DIE MOV-BEFEHLE MOV  SS,XXX UND
;MOV  SP,XXX UNTER 8086/8088 NICHT UNTERBRECHBAR SIND.
        MOV  SS,STACKBASE    ;MARKE ZUM DRIVER-STACK
        MOV     SP,OFFSET TOP_STACK
        PUSH    BX              ;BX UND AX ZUM STACK
        PUSH    AX
        PUSH    BP              ;RETTEN DES BP-REGISTERS
        PUSH    DS              ;RETTEN DER PARAMETERZEIGER
        PUSH    DX
        PUSHF                   ;RETTEN DES RICHTUNGSFLAGS
;AUFRUFEN DES DRIVERS. DS:DX ZEIGT AUF DEN PARAMETERBLOCK
;RUCKKEHR UBER RETF.
        CALLF   DD_DRIVER       ;AUFRUFEN DES DRIVERS
                                ;MIT DS:DX
        POPF                    ;WIEDERHERSTELLEN DES
                                ;RICHTUNGSFLAGS
        POP     DX              ;WIEDERHERSTELLEN DS:DX
        POP     DS
        POP     BP              ;WIEDERHERSTELLEN BP-REG.
        POP     AX              ;WIEDERHERSTELLEN SS:SP
        POP     BX              ;UBER
        MOV     SS,BX           ;BX
        MOV     SP,AX           ;UND AX
        RETF
STACKBASE       DW      SEG TOP_STACK
DD_DRIVER_CODE  CSEG
        EXTERN  DD_DRIVER
STACK           SSEG
                RS      16      ;DIESER MODUL SPEICHERT
                                ;8 WORTE
```

```
;TOP_STACK IST IM LETZTEN LADEMODUL DEFINIERT
                EXTERN    TOP_STACK      ;BYTE
                END
```

Als Hilfe zum Schreiben eines Drivers ist im Anhang 1 ein Driver aufgeführt.
Nachdem der Gerätedriver geschrieben, übersetzt und geladen ist, liegt eine Datei vom Typ .CMD vor. Der Dateityp wird in .SYS geändert:

```
A>REN GIOSXX.SYS=GIOSXX.CMD
```

Um den Driver dem SCP-GX bekannt zu machen, ist sein Name anschließend in die Zuweisungstabelle ASSIGN.SYS einzufügen. Diese Datei ist eine einfache Textdatei mit einem speziellen Format, die die Namen der Driverdateien und die logischen Gerätenummern oder Arbeitsstations-ID enthält, die man den bestimmten Geräten zuordnen will.

Anlage 1 Beispiel fuer einen GIOS - Driver

Die nachfolgenden Ausschnitte aus dem GIOS - Driver für das grafische Bildschirmsystem (GBS) des Arbeitsplatzcomputers A7100 (geschrieben in der Sprache C) sollen als Unterstützung bei der Entwicklung eigener GIOS - Driver dienen. (siehe auch Anhang 4). Anhang 1 enthält nur die zur Demonstration der Struktur eines GIOS - Drivers wesentlichen Routinen. Zum besseren Verständnis werden einige der benutzten, jedoch nicht im Anhang enthaltenen Unterprogramme kurz beschrieben:

blackset	- Schreibmodus des KGS auf Löschen einstellen, falls die Farbe "Schwarz" gefordert war.
blackreset	- Löschmodus wieder rücksetzen.
clearscreen	- Alpha- und Grafikbildschirm löschen.
fillarea	- Ausfüllen eines Polygons in der durch fillstind und fillintst bestimmten Art.
flush	- Grafikkommandopuffer an den KGS übertragen.
ichoice	- Choice - Eingabe
ilocator	- Locator - Eingabe
istring	- Text - Eingabe
itoa	- Konvertierung integer --> ASCII
kgsimode	- KGS - Eingabemodus einstellen (Locator, Choice, String)
loccntrl	- Locator sichtbar bzw. unsichtbar machen.
setlinetyp	- Attribute fuer POLYLINE setzen.
setloc	- Locator auf eine bestimmte Position setzen.
setmarkertyp	- Attribute fuer POLYMARKER setzen.
trf	- Übertragen von X - und Y - Koordinate eines Punktes von einem Quell- nach einem Zielbereich
wodb	- Ausgabe eines Kommandos in den Grafikkommandopuffer.
wesc	- Ausgabe einer ESC-Folge an den KGS.
wtxt	- Text fuer ein Grafik - Textkommando in den Grafikkommandopuffer übertragen.

```
/************************************************/
/* Function: KGS-Driver for SCP 1700 graphics   */
/************************************************/

/*Pointer to graphic parameter block */
extern int **_PARM_POI;
int *contrl;  /*pointer to the control array*/
int *intin;   /*pointer to the input parameter array*/
int *ptsin;   /*pointer to the input point coordinate array*/
int *intout;  /*pointer to the output parameter array*/
int *ptsout;  /*pointer to the output point coordinate array*/
int ilocmd;   /*locator input mode*/
int ichcmd;   /*choice input mode*/
int istrmd;   /*string input mode*/

char linewidth; /*LS for polyline*/
char linetyp;   /*LT*/
char plinec;    /*video index for polyline*/
char markertyp;  /*MT*/
char pmarkerc;   /*video index for polymarker*/
 char  textc;     /*video index for text*/
 char fillintst;  /*FA*/
char fillstind;  /*I*/
char fillc;      /*video index for fill area*/
char writmd;   /*ST*/
```

```
char initmd;      /*IM*/
char colordef[8]={0,2,3,2,1,2,2,2};/* black ->S=0
                  red   ->S=2
                  green ->S=3
                  blue  ->S=1 */
char cidtable[8]={0,1,2,4,7,7,7,7};
#define REQUEST 1
#define X_LIMITS 639
#define Y_LIMITS 399
#define COLORS 8
/*workstation description table*/
int wdt[45]={X_LIMITS, /*screen width in pixels*/
                       Y_LIMITS, /*screen hight in pixels*/
        1,       /*device coordinates units flag*/
          344,   /*width of one pixel*/
          344,   /*higth of one pixel*/
          1,     /*character hights*/
          5,     /*line types*/
          7,     /*line width*/
          28,    /*marker types*/
          1,     /*marker hights*/
          1,     /*fonds*/
          5,     /*pattern*/
          8,     /*hatch styles*/
          4,     /*number of colors*/
          3,     /*number of GDP's*/
          1,     /*BAR*/
          2,        /*ARC*/
          4,     /*circle*/
          -1,
          -1,
          -1,
          -1,
          -1,
          -1,
          -1,
          3,     /*fill area*/
          0,     /*polyline*/
          0,
          -1,
          -1,
          -1,
          -1,
          -1,
          -1,
          -1,
          1,     /*color yes*/
          0,     /*text rotation no*/
          1,     /*fill area yes*/
          1,     /*pixel operation flag*/
          COLORS,/*colors*/
          1,     /*locator device*/
          0,     /*valuator device*/
          1,     /*choice device*/
          1,     /*string device*/
          2};    /*work station type*/
 /*additional work station parameters*/
 int awdt[12]={ 0,
        9,       /*min. char hight*/
        0,
```

```
        9,      /*max. char hight*/
        1,      /*min. line width*/
        0,
        7,      /*max. line width*/
        0,
        0,
        9,      /*min. marker hight*/
        0,
        9};     /*max. marker hight*/
int __BDOS(); /*SCP 1700 call*/

/* get color index */
oid(index,video)
 int index;char *video;
 {if(index>COLORS-1)index=1;
   else if(index<0)index=0;
  *video=colordef[cidtable[index]];
  return(index);
 };

main()
{
 int opcode,i,j,k,*pi;
 char h[6],*ph,*point;
 contrl=*_PARM_POI;
 intin=*(_PARM_POI+1);
 ptsin=*(_PARM_POI+2);
 intout=*(_PARM_POI+3);
 ptsout=*(_PARM_POI+4);

 opcode=*contrl;
 *(contrl+2)=0; /*ptsout count = 0*/

 switch (opcode)
   {case 1:     /*open work station*/
    openws(intout,ptsout);
    break;
    case 2:     /*close workstation*/
    h[0]=4;h[1]=0;wodb(2,h);
    break;
    case 3:     /*clear work station*/
    clearscreen();
    break;
    case 4:     /*update work station*/
    flush();
    break;
    case 5:     /*Escape*/
    escape();
    break;
    case 6:     /*polyline*/
    blackset(plinec);
    polyline(*(contrl+1),ptsin);
    blackreset(plinec);
    break;
    case 7:     /*polymarker*/
    blackset(pmarkerc);
    h[0]=18;ph=&h[1];
    for(i=0;i<=*(contrl+1)-1;++i)
     {point=(char *)(ptsin+2*i);
      trf(ph,point);wodb(5,h);
```

```
 };
blackreset(pmarkerc);
break;
case 8:      /*text*/
blackset(textc);
h[0]=19;ph=&h[1];point=(char *)ptsin;
trf(ph,point);
if((h[5]=textc)==0) h[5]=3;
wcdb(6,h);
wtxt(*(contrl+3),intin);
blackreset(textc);
break;
case 9:      /*fill a polygon*/
fillpoly(*(contrl+1),ptsin);
break;
case 10:     /*cell array*/
cell();
break;
case 11:     /*GDP*/
switch (*(contrl+5))
{case 1:bar(ptsin);
    break;
     case 2:arc(ptsin);
    break;
 case 4:circle(ptsin);
    break;
};
break;
case 12:     /*set character hight*/
*(contrl+2)=2;
pi=ptsout;
*pi++=7;*pi++=9;
*pi++=8;*pi=9;
break;
case 13:     /*set text direction*/
*intout=0;
break;
case 14:     /*specify color representation*/
i=0;
if(*(intin+1) != 0)i=i|1;
if(*(intin+2) != 0)i=i|2;
if(*(intin+3) != 0)i=i|4;
cidtable[*intin]=i;
break;
case 15:     /*set polyline line type*/
if(((linetyp=*intin)>5)||(linetyp<1))linetyp=1;
*intout=linetyp;
setlinetyp(plinec);
break;
case 16:     /*set polyline line width*/
if((linewidth=*ptsin)>7) linewidth=7;
 else if(linewidth<1) linewidth=1;
*(contrl+2)=1;
*ptsout=linewidth;
*(ptsout+1)=0;
setlinetyp(plinec);
break;
case 17:     /*set polyline color index*/
*intout=cid(*intin,&plinec);
setlinetyp(plinec);
```

```
    break;
    case 18:    /*set polymarker typ*/
    if(((markertyp=*intin)>28)||(markertyp<1))markertyp=3;
    *intout=markertyp;
    setmarkertyp();
    break;
    case 19:    /*set polymarker scale*/
    *(contrl+2)=1;
    *(ptsout+1)=0;
    *ptsout=9;
    break;
    case 20:    /*set polymarker color index*/
    *intout=cid(*intin,&pmarkerc);
    setmarkertyp();
    break;
    case 21:    /*set text fond*/
    *intout=1;
    break;
    case 22:    /*set color index for text*/
    *intout=cid(*intin,&textc);
    break;
    case 23:    /*set fill interior style*/
    if(((fillintst=*intin)>3)||(fillintst<0))fillintst=0;
    *intout=fillintst;
    break;
    case 24:    /*set fill style index*/
    if(((fillstind=*intin)>8)||(fillstind<1))fillstind=1;
    if(fillintst==2){if(fillstind>5)fillstind=1;goto end24;};
    if(fillstind==1)fillstind=2;
    else if(fillstind==2)fillstind=1;
end24:
    *intout=fillstind;
    break;
    case 25:    /*set fill color index*/
    *intout=cid(*intin,&fillc);
    break;
    case 26:    /*inquire color representation*/
    *intout=cid(*intin,&i);
    i=cidtable[*intin];
    *(intout+1)=*(intout+2)=*(intout+3)=0;
    if((i & 1)>0) *(intout+1)=1000;
    if((i & 2)>0) *(intout+2)=1000;
    if((i & 4)>0) *(intout+3)=1000;
    break;
    case 27:    /*inquire cell array*/
    break;
    case 28:    /*input locator*/
    kgsimode(1);
    if(ilocmd==1){ setloc(ptsin);
    loccntrl(0x05);};
    *(contrl+2)=1;
    *(contrl+4)=ilocator(ilocmd,ptsout);
    loccntrl(0x04);
    kgsimode(0);
    break;
    case 29:    /*input valuator*/
    break;
    case 30:    /*input choice*/
    kgsimode(4);
    if((*(contrl+4)=ichoice(ichcmd,intout))==0)
```

```
      *intout=*(intin+1);
    kgsimode(0);
    break;
    case 31:    /*input string*/
    kgsimode(32);
    *(contrl+4)=istring(istrmd,intout,*(intin+2),*(intin+1));
    kgsimode(0);
    break;
    case 32:    /*set writing mode*/
    *intout=*intin;
    if(*intin==4) writmd=0;
    else if(*intin==3){writmd=1;*intout=2;}
    else if(*intin==2)writmd=1;
    else {writmd=2;*intout=1;};
    h[0]=23;h[1]=writmd;wcdb(2,h);
    break;
    case 33:    /*set input mode*/
    *intout=1;  /*REQUEST only*/
    break;
 }; /* end case*/
 flush();
 }

/*init kgs*/
initkgs()
 {char h[5],*ph;
  ph=h;
  *ph++=40;*ph++=linetyp;*ph++=plinec;*ph++=markertyp;
  *ph++=pmarkerc;wcdb(5,h);
  writmd=2;
  h[0]=23;h[1]=2;wcdb(2,h);
 };

/*open workstation*/
openws(hintout,hptsout)
 int *hintout,*hptsout;
 {int i,*p;
  *(contrl+2)=6;
  *(contrl+4)=45;
  linewidth=1;
  if(((linetyp=*(intin+1))>5)||(linetyp<1)) linetyp=1;
  cid(*(intin+2),&plinec);
  if(((markertyp=*(intin+3))>28)||(markertyp<1)) markertyp=3;
  cid(*(intin+6),&pmarkerc);
  cid(*(intin+6),&texto);
  if(((fillintst=*(intin+7))>3)||(fillintst<1)) fillintst=0;
  if(((fillstind=*(intin+8))>8)||(fillstind<1)) fillstind=1;
  if(fillstind==1)fillstind=2;else if(fillstind==2)fillstind=1;
  cid(*(intin+9),&fillc);
  ilocmd=ichcmd=istrmd=1;
  p=wdt;
  for(i=0;i<45;++i) *hintout++=*p++;
  p=awdt;
  for(i=0;i<12;++i) *hptsout++=*p++;
  initkgs();
 };

/* Escape functions*/
escape()
 {char h[7],*p;int i,*pi;
```

```
   switch(*(contrl+5))
   {/*Inquire addressable cells*/
    case 1:*intout=24;*(intout+1)=80;
           break;
    /*enter graphics mode*/
    case 2:h[0]=4;h[1]=26;wodb(2,h);
           break;
    /*exit graphics mode*/
    case 3:h[0]=4;h[1]=0;wodb(2,h);
           break;
    case 4:wesc("A");break;       /*cursor up*/
    case 5:wesc("B");break;       /*cursor down*/
    case 6:wesc("C");break;       /*cursor right*/
    case 7:wesc("D");break;       /*cursor left*/
    case 8:wesc(";H");break;      /*home*/
    case 9:wesc("J");break;       /*erase to end*/
    case 10:wesc("K");break;      /*erase to end of line*/
    case 11:itoa(*intin,h);       /*direct cursor address*/
            h[2]=';';itoa(*(intin+1),&h[3]);
            h[5]='H';h[6]=' ';
            wesc(h);break;
    case 12:pi=intin;
        for(i=0;i<*(contrl+3);++i)  /*output text*/
            kgsout(*pi++);
            break;
    case 13:wesc("7m");break;     /*reverse on*/
    case 14:wesc("27m");break;    /*reverse off*/
    case 15:wesc("6n");           /*inquire current cursor address*/
            while(kgsin() !='[');
            *intout=(kgsin()-48)*10+kgsin()-48;
            kgsin();
            *(intout+1)=(kgsin()-48)*10+kgsin()-48;
            kgsin();*(contrl+4)=2;break;
    case 16:*intout=0;break;      /*inquire table status*/
    case 17:break;/*hard copy*/
    case 18:setloc(ptsin);              /*place cursor at location*/
     locontrl(0x05);
     break;
    case 19:locontrl(0x04);    /*remove cursor*/
   }

/* draw polyline*/
polyline(nr,coord)
 int nr,*coord;
 {char h[6],*ph,*point;
  int i;
  h[0]=16;h[1]=1;
  ph=&h[2];point=(char *)coord;
  trf(ph,point);wodb(6,h);
  h[0]=17;ph=&h[1];
  for(i=1;i<nr;++i)

   {point=(char *)(coord+2*i);
    trf(ph,point);wodb(5,h);
   };
 };

/*fill a polygon*/
fillpoly(nr,coord)
 int nr,*coord;
```

```
 {char h[5],*ph,*point;
  if(fillc != plinec)setlinetyp(fillc);
  blackset(fillc);
  polyline(nr,coord);
  /*connect to first point*/
  h[0]=17;ph=&h[1];point=(char *)coord;
  trf(ph,point);wcdb(5,h);
  if(fillc != plinec)setlinetyp(plinec);
   fillarea(nr,coord);
  blackreset(fillc);
 };
/*makes rect angle from given lower left and upper right corner*/
barpoints(dest,src)
 int *src,*dest;
{
  *dest++=*src;
  *dest++=*(src+1);
  *dest++=*src;
  *dest++=*(src+3);
  *dest++=*(src+2);
  *dest++=*(src+3);
  *dest++=*(src+2);
  *dest  =*(src+1);
};

/*GDP bar*/
bar(coord)
 int *coord;
{int hcoord[8];
  barpoints(hcoord,coord);
  fillpoly(4,hcoord);
 };

/* GDP arc */
arc(coord)
 int *coord;
 {char h[15],*ph,*point;
  int i;
  blackset(plinec);
  ph=&h[1];point=(char *)coord;
  h[0]=41;
  for(i=1;i<15;++i) *ph++ = *point++;
  wcdb(15,h);
  blackreset(plinec);
 };
/*GDP circle*/
circle(coord)
 int *coord;
{int hcoord[8];
  if(*(coord+1)+*(coord+4)>wdt[1])return;
  if(*(coord+1)-*(coord+4)<0)return;
  trf(hcoord,coord);
  hcoord[2]=*coord+ *(coord+4);
  hcoord[3]=*(coord+1);
  hcoord[4]=hcoord[2];
  hcoord[5]=hcoord[3]-1;
  trf(&hcoord[6],coord+4);
  arc(hcoord);
};
```

Anlage 2 Virtuelles Geräteinterface (VGI)

Dieser Abschnitt enthält die Spezifikation des Virtuellen Geräteinterface. VGI definiert die Schnittstelle vom Anwenderprogramm zum GDOS, dem geräteunabhängigen Teil des SCP-GX.

Eingabeparameter:

contrl(1) - Operationscode der Gerätefunktion
contrl(2) - Anzahl der Punkte im Bereich ptsin
Jeder Punkt besteht aus einem X,Y-Koordinatenpaar, so daß die Länge dieses Bereiches zweimal so groß ist wie die Anzahl der angegebenen Punkte.
contrl(4) - Länge des Integer-Bereiches intin
contrl(6-n)- operationscodeabhängige Information

intin - Bereich der Integer-Eingabeparameter, Länge in contrl(4)
ptsin - Bereich der Koordinaten der Eingabepunkte

Ausgabeparameter:

contrl(3) - Anzahl der Punkte im Bereich ptsout
Jeder Punkt besteht aus einem X,Y-Koordinatenpaar, so daß die Länge dieses Bereiches zweimal so groß ist wie die Anzahl der angegebenen Punkte.
contrl(5) - Länge des Integerbereiches intout
contrl(6-n)- operationscodeabhängige Information

intout - Bereich der Integer-Ausgabeparameter
ptsout - Bereich der Koordinaten der Ausgabepunkte

Bemerkungen:

Alle Daten, die dem Gerätedriver übergeben werden, werden als 2-Byte-Integerzahlen angenommen. Alle Koordinaten, die dem SCP-GX übergeben werden, sind Normalisierte Koordinaten (0-32767 entlang jeder Achse). Diese Einheiten werden durch SCP-GX in aktuelle Gerätekoordinaten umgewandelt (z.B. Raster für CRT`s oder Schritte für Plotter/Drucker), so daß alle Koordinaten, die dem Gerätedriver übergeben werden, Geräteeinheiten sind.
Es werden sowohl die Ein- als auch die Ausgabekoordinaten durch SCP-GX konvertiert, und das rufende Programm als auch der Gerätedriver müssen sichern, daß der Eingabepunktzähler (contrl(2)) und der Ausgabepunktzähler (contrl(3)) gesetzt sind. contrl(2) muß von dem rufenden Programm auf 0 gesetzt werden, wenn keine X,Y-Koordinaten an SCP-GX zu übergeben sind. In gleicher Weise muß der Gerätedriver contrl(3) auf 0 setzen, wenn keine X,Y-Koordinaten durch SCP-GX zurückgegeben werden.
Da die Eintragungen über den ganzen Bereich jeder Achse gehen, sind die Koordinatenwerte auf den X- und Y-Achsen der Geräte, die keine quadratische Darstellungsfläche besitzen, unterschiedlich verteilt. Alle Bezüge auf Feldelemente in der folgenden VGI-Spezifikation beziehen sich darauf, daß das erste Feldelement den Index 1 hat.
Die GDOS-Ruffolge ist:
1. Übergabe der Steuerung an GDOS mittels Interrupt 224
2. CX enthält den Funktionscode 0473H

3. DS:DX enthalten einen Zeiger zum GDOS-Parameterblock
   (Adresse wird in DS:DX übergeben)
   PB Doppelwortadresse von contrl
   PB+ 4 Doppelwortadresse von intin
   PB+ 8 Doppelwortadresse von ptsin
   PB+12 Doppelwortadresse von intout
   PB+16 Doppelwortadresse von ptsout

Alle Operationscodes müssen vom Driver akzeptiert werden, unabhängig davon, ob sie eine Aktion veranlassen oder nicht. Diese Operationscodes müssen, wie beschrieben, übergeben und ausgeführt werden. Alle Operationscodes sollten, wenn immer möglich, implementiert werden, da eine vollständige Implementation eine bessere Qualität der Grafik garantiert.
Die Gerätedriver-E/A (d.h., die Verbindung zwischen dem Gerätedriver und dem Gerät über die System- Hardware-Ports) wird über BDOS-Rufe des SCP 1700 wahrgenommen. Die CRT-Geräte werden als CONSOLE-Geräte angenommen. Bei Plottern wird vorausgesetzt, daß sie als AXI:AXO-Geräte angeschlossen sind. Bei Druckern wird vorausgesetzt, daß sie als LIST-Geräte angeschlossen sind..

1. OPEN WORKSTATION

Initialisieren einer grafischen Arbeitsstation

Eingabe:

contrl(1) - Operationscode=1
contrl(2) - 0
contrl(4) - Länge von intin=10
intin - Initialisiert die Standards (Linienart, Farbe, Zeichengröße usw.)
intin(1) - Arbeitsstations-Identifier(d.h. Gerätedriver-ID) Dieser Wert wird benutzt, um zu bestimmen, welcher Gerätedriver dynamisch in den Speicher geladen wird.
intin(2) - Linientyp
intin(3) - Polyline-Farbindex
intin(4) - Markertyp
intin(5) - Polymarker-Farbindex
intin(6) - Textart
intin(7) - Text-Farbindex
intin(8) - Füllart für Füllbereiche
intin(9) - Füllindex für Füllbereiche
intin(10) - Füllfarbe für Füllbereiche

Ausgabe:

contrl(3) - Anzahl der Ausgabepunkte=6
contrl(5) - Länge von intout=45
intout(1) - Maximal adressierbare Breite des Bildschirms/Plotters in Rastern/Schritten mit Startpunkt=0 (eine Auflösung von 640 bedeutet einen adressierbaren Bereich von 0-639, d.h. intout(1)=639)
intout(2) - Maximal adressierbare Höhe des Bildschirms/Plotters in Rastern/Schritten mit Startpunkt=0 (eine Auflösung von 400 bedeutet einen adressierbarer Bereich von 0-399, so ist intout(2)=399)

intout(3) - Gerätekoordinateneinheitenflag
0= Gerät ist tauglich für die Erstellung maßstabsgerechter Bilder (typisch sind Plotter und Drukker)
1= Gerät ist nicht tauglich für maßstabsgerechter Bilder (CRT's)
intout(4) - Breite eines Pixels (Plotterschrittes...) in Mikrometern
intout(5) - Höhe eines Pixels (Plotterschrittes...) in Mikrometern
intout(6) - Anzahl der Zeichenhöhen (0=kontinuierlich einstellbar)
intout(7) - Anzahl der Linientypen
intout(8) - Anzahl der Linienstärken
intout(9) - Anzahl der Markertypen
intout(10) - Anzahl der Markergrößen
intout(11) - Anzahl der Textarten
intout(12) - Anzahl der Muster
intout(13) - Anzahl der Schraffurarten
intout(14) - Anzahl der vorbestimmten Farben
(muß mindestens 2 sein, auch für einfarbige Geräte)
Es ist die Anzahl der Farben, die gleichzeitig am Gerät angezeigt werden können.
intout(15) - Anzahl der verallgemeinerten Darstellungselemente (GDPs)
intout(16) - intout(25) - Liste der GDPs
(bis zu 10 sind erlaubt)
-1 GDP existiert nicht
intout(26) - intout(35) - Attribute setzen, die mit jedem GDP verbunden sind.
-1 GDP existiert nicht
0 Polyline
1 Polymarker
2 Text
3 Füllbereich
4 nichts
intout(36) - Farbmöglichkeit
0 nein
1 ja
intout(37) - Textdrehungsmöglichkeit
0 nein
1 ja
intout(38) - Füllbereichsmöglichkeit
0 nein
1 ja
intout(39) - Pixeloperationsmöglichkeit
0 nein
1 ja
intout(40) - Zahl der verfügbaren Farben
0 stetiges Gerät
2 einfarbig (schwarz und weiß)
>2 Zahl der verfügbaren Farben
intout(41) - Zahl der verfügbaren Locator-Geräte
intout(42) - Zahl der verfügbaren Valuator-Geräte
intout(43) - Zahl der verfügbaren Choice-Geräte
intout(44) - Zahl der verfügbaren String-Geräte
intout(45) - Type der Arbeitsstation
0 nur Ausgabe
1 nur Eingabe
2 Eingabe/Ausgabe
3 Geräteunabhängige Segmentspeicherung

```
             4  GKS- Metafile-Ausgabe
ptsout(1)  - 0
ptsout(2)  - Minimale Zeichengröße in Geräteeinheiten
ptsout(3)  - 0
ptsout(4)  - Maximale Zeichengröße in Geräteeinheiten
ptsout(5)  - Minimale Linienstärke in Geräteeinheiten
ptsout(6)  - 0
ptsout(7)  - Maximale Linienstärke in Geräteeinheiten
ptsout(8)  - 0
ptsout(9)  - 0
ptsout(10) - Minimale Markergröße in Geräteeinheiten
ptsout(11) - 0
ptsout(12) - Maximale Markergröße in Geräteeinheiten
```

Die Standardfarbtabelle sollte für ein einfarbiges Gerät und ein Farbgerät unterschiedlich aufgestellt werden.

Einfarbig:

Index	Farbe
0	Schwarz
1	Weiß

Farbig:

Index	Farbe
0	Schwarz
1	Rot
2	Grün
3	Blau
4	Cyan
5	Gelb
6	Magenta
7	Weiß
8-n	Weiß

Andere Standardwerte sollten während der Initialisierung durch den Driver gesetzt werden:
Zeichengröße = minimale Zeichengröße
Zeichenrichtungsvektor = 90 Grad in Uhrzeigerrichtung
von der rechten Waagerechten
(0 Grad-Drehung)
Linienstärke = 1 Geräteeinheit (Raster, Plotterschritt)
Markergröße = minimale Markergröße
Schreibmodus = Ersetzen
Eingabemodus = Request für alle Eingabeklassen
(Locator, Valuator, Choice, String)

Beschreibung:

Die Operation OPEN WORKSTATION veranlaßt, daß ein Grafik- Gerät zum Standardgerät für ein Nutzerprogramm erklärt wird. Das Gerät wird mit den Parametern im Eingabebereich initialisiert und Informationen über das Gerät werden an das GDOS zurückgesendet.

2. CLOSE WORKSTATION

Beenden aller grafischen Ausgaben zu dieser Arbeitsstation

Eingabe:

contrl(1) - Operationscode=2
contrl(2) - 0

Ausgabe:

contrl(3) - 0

Beschreibung:

Die Operation CLOSE WORKSTATION beendet die eigentliche grafische Operation und verhindert irgendwelche weiteren Ausgaben zu dem Gerät.

3. CLEAR WORKSTATION

Löschen des Bildschirmes bei CRT-Geräten, Seitenvorschub bei Druckern oder neues Blatt bei Plottern

Eingabe:

contrl(1) - Operationscode=3
contrl(2) - 0

Ausgabe:

contrl(3) - 0

Beschreibung:

Die Operation CLEAR WORKSTATION veranlaßt CRT-Bildschirme zu löschen und bei Hardcopy-Geräten einen Seitenvorschub durchzuführen. Bei Plottern, ohne Seitenvorschub, wird der Operator veranlaßt eine neue Seite einzulegen.

4. UPDATE WORKSTATION

Anzeigen aller noch anstehenden Grafiken an der Arbeitsstation

Eingabe:

contrl(1) - Operationscode=4
contrl(2) - 0

Ausgabe:

contrl(3) - 0

Beschreibung:

Die Operation UPDATE WORKSTATION veranlaßt, daß alle anstehenden grafischen Kommandos, die in der Warteschlange stehen, unverzüglich ausgeführt werden.

5. ESCAPE

Ausführen einer gerätespezifischen Operation

Eingabe:

contrl(1) - Operationscode=5
contrl(2) - Anzahl der Eingabepunkte
contrl(4) - Anzahl der Eingabeparameter
contrl(6) - Funktionsidentifikator
1=Abfragen der adressierbaren Zeichenpositionen
2=Einstellen des Grafik-Modus
3=Aufheben des Grafik-Modus
4=Cursor um eine Reihe nach oben
5=Cursor um eine Reihe nach unten
6=Cursor um eine Spalte nach rechts
7=Cursor um eine Spalte nach links
8=Cursor zur Ausgangsstellung
9=Löschen bis zum Bildschirmende
10=Löschen bis zum Zeilenende
11=Direkte Adressierung des Cursors
12=Textausgabe ab augenblicklicher Cursorposition
13=Einschalten des Inversmodus
14=Einschalten des Normalmodus
15=Abfragen der aktuellen Cursoradresse
16=Abfragen des Status des Grafik-Tabletts
17=Hardcopy
18=Bewegen des Cursors zur angegebenen Position
19=Cursor wird nicht angezeigt
20-50=Reserviert für spätere Erweiterungen
51-100=Ungenutzt (und verfügbar)
intin - Funktionsabhängige Information
(wird auf den folgenden Seiten beschrieben)
ptsin - Bereich der Eingabekoordinaten für ESCAPE-Funktion

Ausgabe:

contrl(3) - Anzahl der Ausgabepunkte
contrl(5) - Anzahl der Ausgabeparameter
intout - Bereich der Ausgabeparameter
ptsout - Bereich der Ausgabekoordinaten

Beschreibung:

Die ESCAPE-Operation erlaubt, spezielle Fähigkeiten eines Grafik-Gerätes vom Nutzerprogramm aus zu erreichen. Einige ESCAPE-Funktionen werden vorher festgelegt, und andere können für ein spezielles Gerät definiert werden. Die Parameter, die übergeben werden, sind von der Funktion abhängig, die ausgeführt werden soll.

5.1. INQUIRE ADDRESSABLE CHARACTER CELLS

Zurücksenden der Anzahl der vom Alpha-Cursor adressierbaren Spalten und Reihen

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=1

Ausgabe:

contrl(3) - 0
intout(1) - Anzahl der adressierbaren Zeilen auf dem Bildschirm, Standard sind 24 (-1 zeigt an, daß die Cursoradressierung nicht möglich ist)
intout(2) - Anzahl der adressierbaren Spalten auf dem Bildschirm, Standard sind 80 (-1 zeigt an, daß die Cursoradressierung nicht möglich ist)

Beschreibung:

Diese Operation gibt dem rufenden Programm Informationen über die Anzahl der senkrechten (Zeilen) und waagerechten (Spalten) Positionen zurück, wo der Alpha-Cursor auf dem Bildschirm positioniert werden kann.

5.2. ENTER GRAPHICS MODE

Einstellen des Grafik-Modus

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=2

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation veranlaßt das Grafik-Gerät in den Grafik-Modus zu gehen, wenn er verschieden vom Alpha-Modus ist. Sie wird zum Beenden des Alpha-Cursor-Adressierungsmodus und zum eigentlichen Übergang vom Alpha- zum Grafik- Modus verwendet.

5.3. EXIT GRAPHICS MODE

Beenden des Grafik-Modus

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=3

Ausgabe:

contrl(3) - 0

Beschreibung:

Die Operation veranlaßt das Grafik-Gerät den Grafik- Modus zu beenden, wenn er verschieden vom Alpha-Modus ist. Sie wird verwendet, um in den Alpha-Cursor-Adressierungsmodus einzuschalten und vom Grafik- zum Alpha-Modus überzugehen.

5.4. CURSOR UP

Bewegen des Cursors um eine Zeile nach oben,
ohne die horizontale Position zu verändern

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=4

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation bewegt den Alpha-Cursor um eine Zeile nach oben, ohne die horizontale Position zu verändern. Befindet sich der Cursor bereits am oberen Rand, erfolgt keine Reaktion.

5.5. CURSOR DOWN

Bewegen des Cursors um eine Zeile nach unten,
ohne die horizontale Position zu verändern

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=5

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation bewegt den Alpha-Cursor um eine Zeile nach unten, ohne die horizontale Position zu verändern. Befindet sich der Cursor bereits am unteren Rand, erfolgt keine Reaktion.

5.6. CURSOR RIGHT

Bewegen des Cursors um eine Spalte nach rechts, ohne die vertikale Position zu verändern

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=6

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation bewegt den Alpha-Cursor um eine Spalte nach rechts, ohne die vertikale Position zu verändern. Wenn sich der Cursor bereits am rechten Rand befindet, zeigt sich keine Wirkung.

5.7. CURSOR LEFT

Bewegen des Cursors um eine Spalte nach links, ohne die vertikale Position zu verändern

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=7

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation bewegt den Cursor um eine Spalte nach links, ohne die vertikale Position zu verändern. Befindet sich der Cursor bereits am linken Rand, zeigt sich keine Wirkung.

5.8. HOME CURSOR

Bewegen des Cursors zur Ausgangsstellung

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=8

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation bewirkt, den Cursor in die Ausgangsposition zu bewegen (gewöhnlich die linke obere Ecke des CRT-Bildschirmes).

5.9. ERASE TO END OF SCREEN

Löschen von der aktuellen Cursorposition bis zum Bildschirmende

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=9

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation löscht die Anzeigefläche von der aktuellen Cursorposition bis zum Bildschirmende. Die aktuelle Cursorposition wird nicht verändert.

5.10. ERASE TO END OF SCREEN

Löschen von der augenblicklichen Cursorposition bis zum Zeilenende

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=10

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation löscht die Anzeigefläche von der aktuellen Cursorposition bis zum Ende der aktuellen Zeile. Die aktuelle Cursorposition verändert sich nicht.

5.11. DIRECT CURSOR ADDRESS

Bewegen des Cursors zur spezifizierten Zeile und Spalte

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=11
intin(1) - Zeilennummer
intin(2) - Spaltennummer

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation bewegt den Alpha-Cursor direkt zur angegebenen Zeilen- und Spaltenadresse auf der Anzeigefläche.

5.12. OUTPUT CURSOR ADDRESSABLE TEXT

Textausgabe ab der augenblicklichen Cursorposition

Eingabe:

contrl(2) - 0
contrl(4) - Anzahl der Zeichen in der Zeichenfolge
contrl(6) - Funktions-ID=12
intin - Textfolge in ASCII-Zeichen (1 Zeichen pro Wort)

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation gibt den Text ab der augenblicklichen Cursorposition aus.

5.13. REVERSE VIDEO ON

Einschalten des Inversmodus

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=13

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation veranlaßt, daß der nachfolgende Text im Invers-modus angezeigt wird; d.h., die Zeichen sind dunkel auf einem hellen Hintergrund.

5.14. REVERSE VIDEO OFF

Ausschalten des Inversmodus

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=14

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation veranlaßt, daß der gesamte nachfolgende Text im Normalmodus angezeigt wird; d.h., die Zeichen sind hell auf dunklem Hintergrund.

5.15. INQUIRE CURRENT CURSOR ADDRESS

Abfragen der augenblicklichen Cursorposition

Ausgabe:

contrl(3) - 0
intout(1) - Zeilennummer (1-Anzahl der Zeilen)
intout(2) - Spaltennummer (1-Anzahl der Spalten)

Beschreibung:

Diese Operation sendet die augenblickliche Position des Alpha-Cursors in Zeilen und Spalten-Koordinaten zurück.

5.16. INQUIRE TABLET STATUS

Zurücksenden des Tablett-Status

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=16

Ausgabe:

contrl(3) - 0
intout(1) - Tablett-Status
0=Tablett nicht verfügbar
1=Tablett verfügbar

Beschreibung:

Diese Operation zeigt an, ob ein Grafik-Tablett mit der Arbeitsstation verbunden ist.

5.17. HARDCOPY

Generiere Hardcopy

Eingabe:

contrl(2) - 0
contrl(6) - Funktions-ID=17

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation veranlaßt das Gerät, eine Hardcopy zu erzeugen. Diese Funktion ist sehr gerätespezifisch und kann das Kopieren des Bildschirmes auf einen Drucker oder einen anderen zugewiesenen Hardcopy-Gerät zur Folge haben.

5.18. PLACE CURSOR AT LOCATION

Bewegen des Cursors zur angegebenen Stelle

Eingabe:

```
contrl(2) - 2
contrl(6) - Funktions-ID=18
ptsin(1)  - X-Koordinate der Stelle, wohin der Cursor
            gesetzt werden soll
ptsin(2)  - Y-Koordinate der Stelle, wohin der Cursor
            gesetzt werden soll
```

Ausgabe:

```
contrl(3) - 0
```

Beschreibung:

Setzen des Cursors/Markers auf die angegebene Stelle. Dieser ist geräteabhängig und kann ein Unterstrich, Block usw. sein.

5.19. REMOVE CURSOR

Ausblenden des Cursor/Marker

Eingabe:

```
contrl(2) - 0
contrl(6) - Funktions-ID=19
```

Ausgabe:

```
contrl(3) - 0
```

Beschreibung:

Diese Operation macht den Cursor auf dem Bildschirm unsichtbar.

6. POLYLINE

Ausgabe Polyline auf dem Gerät

Eingabe:

```
contrl(1) - Operationscode=6
contrl(2) - Anzahl der Punkte (X/Y-Paare) in der Polyline (n)
ptsin     - Bereich der Koordinaten der Polyline in normalisierten Koordinaten
            ptsin(1)    - X-Koordinate des ersten Punktes
            ptsin(2)    - Y-Koordinate des ersten Punktes
              .
              .
              .
            ptsin(2n-1) - X-Koordinate des letzten Punktes
            ptsin(2n)   - y-Koordinate des letzten Punktes
```

Ausgabe:

```
contrl(3) - 0
```

Beschreibung:

Diese Operation veranlaßt, daß ein Linienzug am Grafik- Gerät angezeigt wird. Der Anfangspunkt des Linienzuges ist der erste Punkt im Eingabebereich. Benachbarte Punkte in ptsin werden miteinander verbunden. Es muß gesichert werden, daß die Linien die aktuellen Linienattribute besitzen: Farbe, Linienart, Linienstärke.

7. POLYMARKER

Ausgabe von Markersymbolen zum Gerät

Eingabe:

```
contrl(1) - Operationscode=7
contrl(2) - Anzahl der Marker
ptsin     - Bereich der Koordinaten in normalisierten
            Koordinaten (n)
            ptsin(1)    - X-Koordinate des ersten Markers
            ptsin(2)    - Y-Koordinate des ersten Markers
              .
              .
              .
            ptsin(2n-1) - X-Koordinate des letzten Markers
            ptsin(2n)   - Y-Koordinate des letzten Markers
```

Ausgabe:

```
contrl(3) - 0
```

Beschreibung:

Diese Operation veranlaßt, die Markersymbole an den Punkten zu zeichnen, die im Eingabebereich spezifiziert werden. Es muß gesichert werden, daß die Linienart 'solid' angegeben wird, bevor die Marker gezeichnet werden, und daß nach Ausführung der

Operation wieder die vorherige Linienart eingestellt wird. Es muß auch gesichert werden, daß die Marker die aktuellen Markeratributе besitzen: Farbe, Größe, Typ.

8. TEXT

Schreiben von Text zur angegebenen Position

Eingabe:

```
contrl(1) - Operationscode=8
contrl(2) - Anzahl der Punkte=1
contrl(4) - Anzahl der Zeichen in der Textfolge
intin     - Zeichenfolge in ASCII-Zeichen (1 Zeichen pro Wort)
ptsin(1)  - X-Koordinate des Anfangspunktes des Textes in
            normalisierten Koordinaten
ptsin(2)  - Y-Koordinate des Anfangspunktes des Textes in
            normalisierten Koordinaten
```

Ausgabe:

```
contrl(3) - 0
```

Beschreibung:

Diese Operation schreibt Text auf die Anzeigefläche, beginnend bei der Position, die durch die Eingabeparameter angegeben wird. Die X,Y-Koordinaten geben die linke, untere Ecke des Zeichens selbst an, nicht des Zeichenfeldes. Es muß gesichert werden, daß der Text die aktuellen Textattribute besitzt: Farbe, Größe, Schreibrichtung, Textart.

9. FILLED AREA

Füllen eines geschlossenen Polygons

Eingabe:

```
contrl(1) - Operationscode=9
contrl(2) - Anzahl der Punkte im Polygon (n)
ptsin     - Bereich der Koordinaten des Polygons in
            normalisierten Koordinaten
            ptsin(1)    - X-Koordinate des 1. Punktes
            ptsin(2)    - Y-Koordinate des 1. Punktes
              .
              .
              .
            ptsin(2n-1) - X-Koordinate des letzten Punktes
            ptsin(2n)   - Y-Koordinate des letzten Punktes
```

Ausgabe:

```
contrl(3) - 0
```

Beschreibung:

Diese Operation füllt ein Polygon, das im Eingabebereich angegeben ist, entsprechend der aktuellen Füllparameter. Es müssen

die richtige Farbe, die Füllart (leer, voll, gemustert, schraffiert) und der Füllindex eingestellt sein, bevor das Füllen ausgeführt wird. Wenn das Gerät den Bereich nicht füllen kann, muß es wenigstens das Polygon in der aktuellen Füllfarbe zeichnen. Der Gerätedriver muß sichern, daß der Füllbereich durch die Verbindung des ersten mit dem letzten Punkt geschlossen ist.

10. CELL ARRAY

Ausgabe einer Zellmatrix

Eingabe:

```
contrl(1) - Operationscode=10
contrl(2) - 2
contrl(4) - Länge des Farbindexbereiches
contrl(6) - Länge einer Reihe im Farbindexbereich
contrl(7) - Anzahl der Elemente, die in jeder Reihe
            des Farbindexbereiches benutzt werden.
contrl(8) - Anzahl der Reihen im Farbindexbereich
contrl(9) - Pixel-Operation, die ausgeführt werden soll.
            1 - Replace - Ersetzen
            2 - Overstrike - Überschreiben
            3 - Complement (XOR) - Komplement
            4 - Erase - Löschen
intin(1)  - Farbindexbereich (zeilenweise abgespeichert)
ptsin(1)  - X-Koordinate der linken, unteren Ecke in
            normalisierten Koordinaten
ptsin(2)  - Y-Koordinate der linken, unteren Ecke in
            normalisierten Koordinaten
ptsin(3)  - X-Koordinate der rechten, oberen Ecke in
            normalisierten Koordinaten
ptsin(4)  - Y-Koordinate der rechten, oberen Ecke in
            normalisierten Koordinaten
```

Ausgabe:

```
contrl(3) - 0
```

Beschreibung:

Diese Operation veranlaßt das Gerät, einen rechteckigen Bereich zu zeichnen, welcher durch die X,Y-Koordinaten und den Farbindexbereich bestimmt wird.
Die Größe der Zellmatrix wird durch die linke, untere und die rechte, obere X,Y-Koordinate bestimmt. Innerhalb des Rechteckes, das durch diese Punkte definiert wird, bestimmt der Farbindexbereich die Farben für die einzelnen Komponenten der Zellmatrix. Der Farbindexbereich muß entsprechend transformiert werden, um die vollständige Breite und Höhe des Rechteckes zu füllen.
Wenn das Gerät die Zellmatrix nicht ausfüllen kann, muß es wenigstens das Rechteck in der aktuellen Linienfarbe ausgeben.

11. GENERALIZED DRAWING PRIMITIVE (GDP)

Ausgabe eines Verallgemeinerten Darstellungselementes

Eingabe:

```
contrl(1) - Operationscode=11
contrl(2) - Anzahl der Punkte in ptsin
contrl(4) - Länge des Eingabebereichs intin
contrl(6) - Primitiven-ID
            1 - Balken: verwendet Attribute für das Bereichs-
                        füllen (Füllart, Füllindex, Füllfarbe)
            2 - Kreisbogen: verwendet Linienattribute (Farbe,
                            Linienart, Stärke)
            3 - Kreissektor: verwendet Attribute für das
                             Bereichsfüllen (Füllart,
                             Füllindex, Füllfarbe)
            4 - Kreis: verwendet Attribute für das Bereichs-
                       füllen (Füllart, Füllindex, Füllfarbe)
            5 - Drucken grafischer Zeichen
            6-7:  ungenutzt, aber reserviert für spätere
                  Erweiterungen
            8-10: ungenutzt, aber für die Benutzung verfügbar
ptsin     - Bereich der Koordinaten der GDP in normalisierten
            Koordinaten
            ptsin(1)    - X-Koordinate des ersten Punktes
            ptsin(2)    - Y-Koordinate des ersten Punktes
              .
              .
              .
            ptsin(2n-1) - X-Koordinate des letzten Punktes
            ptsin(2n)   - Y-Koordinate des letzten Punktes
            intin       - Daten
```

Balken:

```
contrl(2) - 2 (Anzahl der Punkte)
contrl(6) - 1 (Primitiven-ID)
ptsin(1)  - X-Koordinate der linken, unteren Ecke des Balkens
ptsin(2)  - Y-Koordinate der linken, unteren Ecke des Balkens
ptsin(3)  - X-Koordinate der rechten, oberen Ecke des Balkens
ptsin(4)  - Y-Koordinate der rechten, oberen Ecke des Balkens
```

Kreisbögen und Kreissektoren:

```
contrl(2) - 4 (Anzahl der Punkte)
contrl(6) - 2 (Kreisbogen) oder 3 (Kreissektor)
intin(1)  - Anfangswinkel in Zehntelgraden (0-3600)
intin(2)  - Endwinkel in Zehntelgraden (0-3600)
ptsin(1)  - X-Koordinate des Mittelpunktes des Kreisbogens
ptsin(2)  - Y-Koordinate des Mittelpunktes des Kreisbogens
ptsin(3)  - X-Koordinate des Startpunktes des Kreisbogens
ptsin(4)  - Y-Koordinate des Startpunktes des Kreisbogens
ptsin(5)  - X-Koordinate des Endpunktes des Kreisbogens
ptsin(6)  - Y-Koordinate des Endpunktes des Kreisbogens
ptsin(7)  - Radius
ptsin(8)  - 0
```

Kreis:

contrl(2) - 3 (Anzahl der Punkte)
contrl(6) - 4 (Primitiven-ID)
ptsin(1) - X-Koordinate des Mittelpunktes des Kreises
ptsin(2) - Y-Koordinate des Mittelpunktes des Kreises
ptsin(3) - X-Koordinate eines Punktes auf dem Kreis
ptsin(4) - Y-Koordinate eines Punktes auf dem Kreis
ptsin(5) - Radius
ptsin(6) - 0

Drucken grafischer Zeichen:(für grafische Drucker)

contrl(2) - 1 (Anzahl der Punkte)
contrl(4) - Anzahl der Zeichen zur Ausgabe
contrl(6) - 5
intin - grafische Zeichen zur Ausgabe
ptsin(1) - X-Koordinate des Anfangspunktes des Zeichens
ptsin(2) - Y-Koordinate des Anfangspunktes des Zeichens

Ausgabe:

contrl(3) - 0

Beschreibung:

Die GDP-Operation gestattet es, die Hardware-Möglichkeiten eines grafischen Gerätes auszunutzen. Die Erzeugung spezieller Darstellungselemente, wie Kreisbögen und Kreise, können durch diesen Mechanismus direkt angestoßen werden.
Verschiedene Primitiven-Identifier sind vordefiniert und andere sind für Erweiterungen verfügbar. Steuer- und Datenbereiche sind von der Art der Primitive abhängig.
In einigen GDP's (Kreisbögen, Kreise, Kreissektoren) werden redundante Parameterangaben gefordert. Es brauchen jedoch nur die für das spezielle Gerät notwendigen Angaben benutzt werden.
Alle Winkelangaben setzen voraus, daß sich der Nullpunkt 90 Grad rechts von der Vertikalen befindet, und daß die Winkel entgegen der Uhrzeigerrichtung zunehmen.

12. SET CHARACTER HEIGHT

Festlegen der Zeichenhöhe

Eingabe:

contrl(1) - Operationscode=12
contrl(2) - Anzahl der Punkte=1
ptsin(1) - 0
ptsin(2) - Geforderte Zeichenhöhe in normalisierten Koordinaten

Ausgabe:

contrl(3) - Anzahl der Punkte=2
ptsout(1) - Aktuelle Zeichenbreite in normalisierten Koordinaten
ptsout(2) - Aktuelle Zeichenhöhe in normalisierten Koordinaten

ptsout(3) - Zeichenmatrixbreite in normalisierten Koordinaten
ptsout(4) - Zeichenmatrixhöhe in normalisierten Koordinaten

Beschreibung:

Diese Operation setzt die augenblickliche Zeichenhöhe in normalisierten Koordinaten. Die angegebene Höhe ist die Höhe des Zeichens selbst und nicht die Höhe der Zeichenmatrix. Der Driver sendet sowohl die Größe des Zeichens als auch die der Zeichenmatrix zurück.

13. SET CHARACTER UP VECTOR

Festlegen der Schreibrichtung für Textausgaben

Eingabe:

contrl(1) - Operationscode=13
contrl(2) - 0
intin(1) - geforderter Drehwinkel der Zeichengrundlinie
(in Zehntelgraden 0-3600)
intin(2) - cos(Winkel)*100 (0-100)
intin(3) - sin(Winkel)*100 (0-100)

Ausgabe:

contrl(3) - 0
intout(1) - eingestellter Winkel der Zeichengrundlinie
(in Zehntelgrad)

Beschreibung:

Diese Operation fordert einen Drehwinkel, der in Zehntelgraden für den CHARACTER UP VECTOR gegeben ist, welcher die Basislinie für den nachfolgenden Text angibt. Der Driver gibt den Drehwinkel zurück, der dem geforderten Wert am nahesten kommt. Aus Bequemlichkeit werden redundante Informationen bei der Eingabe verlangt. Es braucht aber nur die Information genutzt zu werden, die als Parameterangabe für das spezielle Gerät am geeignetsten ist. Die Winkelspezifikation setzt voraus, daß die Position 0 rechts von der Senkrechten (Ost beim Kompaß) liegt, und daß die Winkel entgegen der Uhrzeigerrichtung zunehmen.

14. SET COLOR REPRESENTATION

Spezifikation des Farbindexwertes

Eingabe:

contrl(1) - Operationscode=14
contrl(2) - 0
intin(1) - Farbindex
intin(2) - Rot-Farbintensität (in Zehntelprozent 0-1000)
intin(3) - Grün-Farbintensität
intin(4) - Blau-Farbintensität

Ausgabe:

contrl(3) - 0

Beschreibung:

Diese Operation verbindet einen Farbindex mit der Farbe, die in RGB-Einheiten spezifiziert ist. Mindestens zwei Farbindizes sind notwendig. (schwarz und weiß für einfarbig)

15. SET POLYLINE LINETYPE

Setzen des Polyline-Linientyps

Eingabe:

contrl(1) - Operationscode=15
contrl(2) - 0
intin(1) - geforderter Linientyp

Ausgabe:

contrl(3) - 0
intout(1) - ausgewählter Linientyp

Beschreibung:

Diese Operation setzt den Linientyp für die folgenden Polyline-Operationen fest. Die Gesamtzahl der möglichen Linientypen ist geräteabhängig. Es werden jedoch 5 Linienarten gefordert, eine durchgängige plus 4 unterbrochene Linien. Wenn der geforderte Linientyp außerhalb des Bereiches liegt, dann sollte Linientyp 1 (durchgängig) benutzt werden.

16. SET POLYLINE LINE WIDTH

Setzen der Polyline-Linienstärke

Eingabe:

contrl(1) - Operationscode=16
contrl(2) - Anzahl der Eingabepunkte=1
ptsin(1) - geforderte Linienstärke in normalisierten Koordinaten
ptsin(2) - 0

Ausgabe:

contrl(3) - Anzahl der Ausgabepunkte=1
ptsout(1) - ausgewählte Linienstärke in normalisierten Koordinaten
ptsout(2) - 0

Beschreibung:

Diese Operation setzt die Linienstärke für die folgenden Polyline-Operationen. Die Größe wird in normalisierten Koordinaten (NDC) angegeben.

17. SET POLYLINE COLOR INDEX

Festlegen des Polyline-Farbindexes

Eingabe:

contrl(1) - Operationscode=17
contrl(2) - 0
intin(1) - geforderter Farbindex

Ausgabe:

contrl(3) - 0
intout(1) - ausgewählter Farbindex

Beschreibung:

Diese Operation setzt den Farbindex für die folgenden Polyline-Operationen. Die durch den Index bezeichnete Farbe wird durch die SET COLOR REPRESENTATION-Operation bestimmt. Mindestens zwei Farbindizes sind auswählbar. Die Farbindizes bewegen sich zwischen 0 und einem geräteabhängigen Maximum.

18. SET POLYMARKER TYPE

Setzen des Polymarker-Typ

Eingabe:

contrl(1) - Operationscode=18
contrl(2) - 0
intin(1) - geforderter Polymarker-Typ

Ausgabe:

contrl(3) - 0
intout(1) - ausgewählter Polymarker-Typ

Beschreibung:

Diese Operation setzt den Markertyp für die folgenden Polymarker-Operationen. Die Gesamtzahl der verfügbaren Markertypen ist geräteabhängig, es sind jedoch 5 Markertypen festgelegt.

1 .
2 +
3 *
4 o
5 x

Falls der geforderte Markertyp außerhalb des Bereiches liegt, sollte der Typ 3 benutzt werden.

19. SET POLYMARKER SCALE

Festlegen der Größe für Polymarker

Eingabe:

contrl(1) - Operationscode=19
contrl(2) - Anzahl der Eingabepunkte=1
ptsin(1) - 0
ptsin(2) - geforderte Polymarker-Höhe in normaliserten Geräteeinheiten

Ausgabe:

contrl(3) - 0
ptsout(1) - 0
ptsout(2) - ausgewählte Polymarker-Höhe in normaliserten Geräteeinheiten

Beschreibung:

Diese Operation verlangt eine Polymarker-Höhe für die nachfolgenden Polymarker-Operationen. Der Driver gibt die aktuelle ausgewählte Höhe zurück, die der geforderten Höhe am nahesten kommt.

20. SET POLYMARKER COLOR INDEX

Setzen des Polymarker-Farbindex

Eingabe:

contrl(1) - Operationscode=20
contrl(2) - 0
intin(1) - geforderter Polymarker-Farbindex

Ausgabe:

contrl(3) - 0
intout(1) - ausgewählter Polymarker-Farbindex

Beschreibung:

Diese Operation setzt den Farbindex für die nachfolgenden Polymarker-Operationen. Der Wert des Index wird durch die COLOR-Operation (Operationscode 14) festgelegt. Mindestens 2 Farbindizes werden sind auswählbar.

21. SET TEXT FONT

Festlegen der geräteabhängigen Textart

Eingabe:

contrl(1) - Operationscode=21
contrl(2) - 0
intin(1) - geforderte Nummer der geräteabhängigen Textart

Ausgabe:

contrl(3) - 0
intout(1) - ausgewählte geräteabhängige Textart

Beschreibung:

Diese Operation wählt eine Zeichenart für nachfolgende Textoperationen aus. Die Arten sind geräteabhängig und werden von 1 bis zu einem geräteabhängigen Maximum angegeben.

22. SET TEXT COLOR INDEX

Setzen des Text-Farbindexes

Eingabe:

contrl(1) - Operationscode=22
contrl(2) - 0
intin(1) - geforderter Text-Farbindex

Ausgabe:

contrl(3) - 0
intout(1) - ausgewählter Text-Farbindex

Beschreibung:

Diese Operation setzt den Farbindex für nachfolgende Textoperationen. Mindestens zwei Farbindizes sind auswählbar. Die Farbindizes liegen in einem Bereich von 0 bis zu einem geräteabhängigen Maximum.

23. SET FILL INTERIOR STYLE

Setzen der Füllart

Eingabe:

contrl(1) - Operationscode=23
contrl(2) - 0
intin(1) - geforderte Füllart
0 leer
1 voll
2 gemustert
3 schraffiert

Ausgabe:

contrl(3) - 0
intout(1) - ausgewählte Füllart

Beschreibung:

Diese Operation setzt die Füllart, die für die nachfolgenden Polygon-Filloperationen benutzt wird. Falls eine geforderte Füllart nicht verfügbar ist, sollte die Art 0 benutzt werden. Die genutzte Art wird an das rufende Programm zurückgegeben.

24. SET FILL STYLE INDEX

Setze Füllindex

Eingabe:

contrl(1) - Operationscode=24
contrl(2) - 0
intin(1) - geforderter Füllindex für gemustertes oder schraffiertes Füllen

Ausgabe:

contrl(3) - 0
intout(1) - ausgewählter Füllindex für gemustertes oder schraffiertes Füllen

Beschreibung:

Die Auswahl des Index basiert auf der Füllart. Dieser Index hat keinen Effekt, wenn die Füllart entweder "voll" oder "leer" ist. Die Indizes gehen von 1 bis zu einem geräteabhängigen Maximum. Falls der geforderte Index nicht verfügbar ist, sollte der Index 1 genutzt werden. Der Index verweist auf eine Schraffurart, falls die Füllart "schraffiert" ist, oder er verweist auf ein Muster (Sterne, Punkte usw.), falls die Füllart "gemustert" ist. Vereinbarungsgemäß sollte die Schraffur wie folgt ausgefüllt werden:

1 senkrechte Linien
2 horizontale Linien
3 +45 Grad Linien
4 -45 Grad Linien
>4 geräteabhängig

25. SET FILL COLOR INDEX

Setze Füll-Farbindex

Eingabe:

contrl(1) - Operationscode=25
contrl(2) - 0
intin(1) - geforderter Füll-Farbindex

Ausgabe:

contrl(3) - 0
intout(1) - ausgewählter Füll-Farbindex

Beschreibung:

Diese Operation setzt den Farbindex für nachfolgende Polygon-Filloperationen. Der aktuelle RGB-Wert des Farbindexes wird durch die SET COLOR REPRESENTATION-Operation bestimmt. Mindestens zwei Farbindizes werden gefordert. Die Farbindizes liegen im Bereich von 0 bis zu einem geräteabhängigen Maximum.

26. INQUIRE COLOR REPRESENTATION

Zurückgeben der Farbdarstellung

Eingabe:

```
contrl(1) - Operationscode=26
contrl(2) - 0
intin(1)  - geforderter Farbindex
intin(2)  - Flag
            0  Gesetzt (Zurückgeben des geforderten Farbwertes)
            1  Realisiert (Zurückgeben des Farbwertes,
                           der im Gerät realisiert ist)
```

Ausgabe:

```
contrl(3) - 0
intout(1) - Farbindex
intout(2) - Rot-Intensität (in Zehntel Prozent)
intout(3) - Grün-Intensität
intout(4) - Blau-Intensität
```

Beschreibung:

Diese Operation gibt den geforderten oder den aktuellen Wert für einen spezifizierten Farbindex in RGB-Einheiten zurück.

Beachte:
Der Gerätedriver muß Tabellen für die Farbwerte aufstellen, die gefordert wurden und solche für die Farbwerte, die realisiert wurden. In Geräten, die einen kontinuierlichen Farbbereich haben, kann eine dieser Tabellen entfallen.

27. INQUIRE CELL ARRAY

Zurückgeben der Definition einer Zellmatrix

Eingabe:

```
contrl(1) - Operationscode=27
contrl(2) - 2
contrl(4) - Länge des Farbindex-Bereiches
contrl(6) - Länge jeder Reihe im Farbindex-Bereich
contrl(7) - Anzahl der Reihen im Farbindex-Bereich
ptsin(1)  - X-Koordinate der unteren, linken Ecke in
            normalisierten Koordinaten
ptsin(2)  - Y-Koordinate der unteren, linken Ecke in
            normalisierten Koordinaten
ptsin(3)  - X-Koordinate der oberen, rechten Ecke in
            normalisierten Koordinaten
ptsin(4)  - Y-Koordinate der oberen, rechten Ecke in
            normalisierten Koordinaten
```

Ausgabe:

```
contrl(3) - 0
contrl(8) - Anzahl der genutzten Elemente in jeder Reihe
            des Farbindexbereich
contrl(9) - Anzahl der genutzten Reihen im Farbindex-Bereich
```

```
contrl(10)- Fehlerflag
            0  falls keine Fehler
            1  falls ein Farbwert für irgendein Pixel nicht
               bestimmt werden konnte
intout      - Farbindex-Bereich (reihenweise abgespeichert)
              -1 zeigt an, daß für ein einzelnes Pixel der
                 Farbindex nicht bestimmt werden kann
```

Beschreibung:

Diese Operation gibt die Definition der Zellmatrix für die spezifizierte Zelle zurück. Die Farbindex-Werte werden reihenweise zurückgegeben, beginnend vom oberen Eckpunkt des rechteckigen Bereiches nach unten.

28. INPUT LOCATOR

Zurücksenden der Locator-Position

REQUEST-Modus

Eingabe:

```
contrl(1) - Operationscode=28
contrl(2) - Anzahl der Eingabepunkte=1
intin(1)  - Locator-Gerätenummer
            1  Standard-Locatorgerät
            2  Fadenkreuz
            3  grafisches Tablett
            4  Joystick
            5  Lichtstift
            6  Plotter
            7  Maus
            8  Trackball
           >8  arbeitsstationsabhängig
ptsin(1)  - X-Koordinate des Locators in normalisierten
            Koordinaten
ptsin(2)  - Y-Koordinate des Locators in normalisierten
            Koordinaten
```

Ausgabe:

```
contrl(3) - Anzahl der Ausgabepunkte=1
contrl(5) - Länge des intout-Bereiches
            Status: 0  Forderung nicht erfolgreich
                    >0 Forderung erfolgreich
intout(1) - Locator-Eingabe-Abschluß
            Auf der Tastatur wird die Locator-Eingabe durch eine
            spezielle Eingabeende-Taste abgeschlossen. Bei
            Geräten ohne Tastatur (Tablett, Maus usw.) wird die
            Eingabe durch gültige Locator-Terminatoren, beginnend
            mit <Space> (DEZ 32) und von dort ansteigend,
            beendet. Zum Beispiel, falls der Puck auf einem
            Tablett 4 Knöpfe hat, sollte der 1. Knopf ein Space
            als Terminator generieren, der zweite ein <!> (DEZ
            32), der 3. Knopf ein <"> (DEZ 34) und der 4. Knopf
            ein <#> (DEZ 35) generieren.
ptsout(1) - End-X-Koordinate des Locators in normalisierten
            Koordinaten
```

ptsout(2) - End-Y-Koordinate des Locators in normalisierten Koordinaten

SAMPLE-Modus

Eingabe:

contrl(1) - Operationscode=28
contrl(2) - Anzahl der Eingabepunkte=0
intin(1) - Locator-Gerätenummer
1 Standard-Locatorgerät
2 Tastatur
3 grafisches Tablett
4 Joystick
5 Lichtstift
6 Plotter
7 Maus
8 Trackball
>8 arbeitsstationsabhängig

Ausgabe:

contrl(3) - Zahl der Ausgabepunkte
1 Sample erfolgreich
0 Sample nicht erfolgreich
contrl(5) - Länge des Eingabebereich-Status
0 Sample nicht erfolgreich
>0 Sample erfolgreich
ptsout(1) - aktuelle X-Koordinate des Locators in normalisierten Koordinaten
ptsout(2) - aktuelle Y-Koordinate des Locators in normalisierten Koordinaten

Beschreibung:

Diese Operation gibt die Position des spezifizierten Locator-Gerätes in normalisierten Koordinaten zurück.

29. INPUT VALUATOR

Zurücksenden des Valuator-Gerätes

REQUEST-Modus

Eingabe:

contrl(1) - Operationscode=29
contrl(2) - 0
intin(1) - Valuator-Gerätenummer
1 Standard-Valuator-Gerät
intin(2) - Anfangswert

Ausgabe:

contrl(3) - 0
contrl(5) - Länge des intout-Bereiches
Status: 0 Forderung nicht erfolgreich
>0 Forderung erfolgreich

```
intout(1) - Ausgabewert
```

SAMPLE-Modus

Eingabe:

```
contrl(1) - Operationscode=29
contrl(2) - 0
intin(1)  - Valuator-Gerätenummer
            1  Standard-Valuator-Gerät
```

Ausgabe:

```
contrl(3) - 0
contrl(5) - Länge des intout-Bereiches
            Status: 0  Sample nicht erfolgreich
                    >0  Sample erfolgreich
intout(1) - aktueller Valuator-Wert, falls Sample erfolgreich
```

Beschreibung:

Diese Operation gibt den aktuellen Wert des Valuator-Gerätes zurück.

30. INPUT CHOICE

Zurückgeben des Choice-Gerätestatus

REQUEST-Modus

Eingabe:

```
contrl(1) - Operationscode=30
contrl(2) - 0
intin(1)  - Choice-Gerätenummer
            1  Standard-Choice-Gerät
            2  Funktionstaste
            (>2  arbeitsstationsabhängig
              3  grafisches Tablett CHOICE1
              4  grafisches Tablett CHOICE2)
intin(2)  - Anfangs-Choice-Nummer
```

Ausgabe:

```
contrl(3) - 0
contrl(5) - Länge des intout-Bereiches
            Status: 0  Forderung nicht erfolgreich
                    >0  Forderung erfolgreich
intout(1) - Choice-Nummer (z.B. Nummer der gedrückten
            Funktionstaste)
```

SAMPLE-Modus

Eingabe:

```
contrl(1) - Operationscode=30
contrl(2) - 0
```

```
intin(1)  - Choice-Gerätenummer
            1  Standard-Choice-Gerät
            2  Funktionstaste
           >2  arbeitsstationsabhängig
```

Ausgabe:

```
contrl(3) - 0
contrl(5) - Länge des intout-Bereiches
            Status: 0  Sample nicht erfolgreich
                   >0  Sample erfolgreich
intout(1) - Choice-Nummer, falls Sample erfolgreich
```

Beschreibung:

Diese Operation gibt den Choice-Status des angegebenen Choice-Gerätes zurück. Der Bereich der Choice-Nummern ist geräteabhängig.

31. INPUT STRING

Zurücksenden einer Zeichenkette vom angegebenen String-Gerät

REQUEST-Modus

Eingabe:

```
contrl(1) - Operationscode=31
contrl(2) - 0
intin(1)  - String-Gerätenummer
            1  Standard-String-Gerät (Tastatur)
            (>1 arbeitsstationabhängig
              2 Tastatur
              3 grafisches Tablett)
intin(2)  - maximale Stringlänge
intin(3)  - Echo-Modus
            0  kein Echo der Eingabezeichen
            1  Echo der Eingabezeichen
```

Ausgabe:

```
contrl(3) - 0
contrl(5) - Länge der Ausgabenfolge
            0  Forderung nicht erfolgreich
           >0  Forderung erfolgreich
intout    - Ausgabenfolge
```

SAMPLE-Modus

Eingabe:

```
contrl(1) - Operationscode=31
contrl(2) - 0
intin(1)  - String-Gerätenummer
            1  Standard-String-Gerät (Tastatur)
intin(2)  - maximale Stringlänge
intin(3)  - Echomodus
            0  kein Echo der Eingabezeichen
            1  Echo der Eingabezeichen
```

Ausgabe:

```
contrl(3) - 0
contrl(5) - Länge der Ausgabenfolge
            0  Sample nicht erfolgreich
               (Zeichen nicht verfügbar)
           >0  Sample erfolgreich
               (Zeichen verfügbar)
intout    - Ausgabenfolge, falls Sample erfolgreich
```

Beschreibung:

Diese Operation gibt eine Zeichenfolge vom angegebenen Gerät zurück. Das Standardgerät ist die Tastatur.

32. SET WRITING MODE

Schreibmodus setzen

Eingabe:

```
contrl(1) - Operationscode=32
contrl(2) - 0
intin(1)  - Schreibmodus
            1  Ersetzen
            2  Überschreiben
            3  Komplement
            4  Löschen
```

Ausgabe:

```
contrl(3) - 0
intout    - ausgewählter Schreibmodus
```

Beschreibung:

Diese Operation beeinflußt die Art und Weise, wie Pixel für Linien, Füllbereiche, Text usw. auf dem Bildschirm erscheinen.

33. SET INPUT MODE

Eingabemodus setzen

EINGABE:

```
contrl(1) - Operationscode=33
contrl(2) - 0
intin(1)  - logisches Eingabegerät
            1  Locator
            2  Valuator
            3  Choice
            4  String
intin(2)  - Eingabemodus
            1  Request
            2  Sample
```

Ausgabe:

```
contrl(3) - 0
intout    - ausgewählter Eingabemodus
```

Beschreibung:

Diese Operation setzt den Eingabemodus für das angegebene logische Eingabegerät (Locator, Valuator, Choice, String) entweder Request oder Sample. Im Request-Modus wartet der Driver auf eine Eingabe, bevor zurückgekehrt wird. Im Sample-Modus gibt der Driver den aktuellen Status/Location des Eingabegerätes ohne zu warten zurück.

34. INPUT STROKE

Eingabe einer Punktfolge

REQUEST-Modus

Eingabe

```
contrl(1) - Operationscode=34
contrl(2) - Anzahl der Eingabepunkte für STROKE-Anfang
            (0 <= n <= 16)
intin(1)  - Stroke-Gerätenummer
            1 Standard-Stroke-Gerät
            2 Tastatur
            3 grafisches Tablett
intin(2)  - maximale Anzahl von STROKE-Punkten nmax
            (1 < nmax <= 255)
intin(3)  - echotyp
            0 kein Echo
            1 Marker
            2 Verbindungslinien
            3 Marker und Verbindungslinien
ptsin     - Bereich der Koordinaten für den STROKE-Anfang
            ptsin(1)    - X-Koordinate des ersten Punktes
            ptsin(2)    - Y-Koordinate des ersten Punktes
              .
              .
              .
            ptsin(2n-1) - X-Koordinate des letzten Punktes
            ptsin(2n)   - Y-Koordinate des letzten Punktes
```

Ausgabe

```
contrl(3) - Länge des STROKE
            (Anzahl der Punkte im ptsout)
            0 REQUEST nicht erfolgreich
           >0 REQUEST erfolgreich
ptsout    - Bereich der STROKE-Punkte
            (maximal nmax Koordinatenpaare)
```

Beschreibung

Die Funktion liefert eine Punktfolge (STROKE) in normalisierten Koordinaten vom angegebenen Gerät (maximal 255 Punkte).
Ist contrl(2) ungleich 0, werden die in ptsin spezifizierten Punkte als Anfangswerte in der Ausgabepunktfolge übernommen.

Allgemeine Hinweise zur Nutzung von Kommandos als VGI in Anwenderprogrammen

- Die Grafiktreiber für unterschiedliche Grafikgeräte realisieren im allgemeinen ein Subset des VGI. Anwenderprogramme können davon ausgehen, daß folgende Kommandos des VGI realisiert sein müssen:

  1 Open Workstation
  2 Close Workstation
  3 Clear Workstation
  4 Update Workstation
  5 Escape
    Unterfunktion 1
    Unterfunktionen 2 - 12 (nur bei CRT-Geräten)
    Unterfunktion 15 (nur bei CRT-Geräten)
  6 Polyline
  7 Polymarker
  8 Text
  9 Filled area
  10 Cell array
  12 Set charakter height
  14 Set color representation
  15 Set polyline linetyp
  17 Set polyline color index
  18 Set polymarker type
  20 Set polymarker color index
  22 Set text color index
  25 Set fill color index
  26 Inquire color representation
  33 Set input mode (nur erforderlich, wenn Eingabefunktionen verfügbar sind)

  Um festzustellen, welche Kommandos ein Treiber über dieses Subset hinaus realisiert, kann von Anwenderprogrammen die Rückkehrinformation beim Open Workstation-Ruf ausgewertet werden. Für die vom Kombinat Robotron vertriebenen Grafikdriver ist der Funktionsumfang in Anlage 4 beschrieben.

- In Bezug auf Geräte, die keine quadratische Anzeigefläche besitzen, sind die Anmerkungen in Punkt 3.3. zu beachten.

Anlage 3 Begriffserklärungen

Zuweisungstabelle:

Die Zuweisungstabelle verbindet die logischen Gerätenummern, Arbeitsstations-IDs genannt, mit den spezifizierten GIOS-Dateien, so daß innerhalb des Nutzerprogrammes mit den Nummern auf die Geräte Bezug genommen werden kann. Die Zuweisungstabelle befindet sich in einer Textdatei namens ASSIGN.SYS und kann mit einem Texteditor modifiziert werden.

BDOS:

BDOS ist das SCP 1700 Basic Disk Operating System. Es enthält den geräteunabhängigen Teil des SCP 1700. Die geräteabhängigen Teile des SCP 1700 sind im BIOS (Basic I/O-System).

CMD-Datei:

Eine .CMD-Datei identifiziert ausführbare Programmdateien.

DC:

Abkürzung für Gerätekoordinaten (device coordinates)

Koordinaten-Skalierung:

Transformieren von Punkten aus einem Koordinatenraum in einen anderen. Bei Aufruf der SCP-GX-Funktion durch das Anwenderprogramm müssen alle Punktkoordinaten Normalisierte Koordinaten mit Werten zwischen 0 und 32767 angegeben sein. GDOS transformiert diese Koordinaten in Gerätekoordinaten des jeweiligen Grafikgerätes.

Standard-Gerätedriver:

Der größte Driver, der während einer grafischen Darstellung geladen wurde. Es ist immer der erste Driver in der Zuweisungstabelle ASSIGN.SYS.

Gerätedriver:

Ein Gerätedriver ist eine GIOS-Datei, die geräteunabhängige grafische Standardoperationen in spezielle grafische Kommandofolgen für das einzelne Gerät transformiert. Die Gerätedriver für grafische Geräte sind in dem GIOS-Teil (Graphics-I/O-System) des SCP-GX enthalten.

Funktionscode:

Ein Funktionscode ist eine Zahl, die dem Betriebssystem mitteilt, welche Funktion gefordert wird, wenn ein Systemruf abgesetzt wird. Alle grafischen Funktionen benutzen den Funktionscode 0473H. Die spezielle gewünschte grafische Operation wird durch einen Operationscode in der an das GDOS zu übergebenden Parameterliste spezifiziert.

GDOS:

Das Graphics Device Operating System (GDOS) ist der geräteunabhängige Teil des SCP-GX. Es nimmt die Grafikrufe entgegen und ruft GIOS, um die Kommandos zu den grafischen Geräten zu senden.

Verallgemeinertes Darstellungselement (GDP):

Ein spezielles Ausgabeelement zum Ansprechen spezifischer grafischer Fähigkeiten einer Arbeitsstation.

GIOS:

Das Graphics Input Output System (GIOS) ist der geräteabhängige Teil des SCP-GX. GIOS-Dateien sind spezielle Gerätedriver, welche die Verbindung zwischen einem einzelnen Gerät und den Standard-VGI-Konventionen herstellen.

GRAPHICS-Modus:

Der grafische Modus wird eingestellt durch die Ausführung des GRAPHICS-Kommandos vom SCP 1700-Monitor: A>GRAPHICS. Dadurch werden die grafischen Funktionen erlaubt. Der grafische Modus wird durch das Kommando A>GRAPHICS NO oder ein erneutes Booten ausgeschaltet.

SCP-GX:

SCP-GX ist die grafische Erweiterung des Betriebssystems SCP 1700.

Grafikprimitiven:

Grafikprimitiven sind die Grundelemente der grafischen Operationen, die durch das SCP-GX ausgeführt werden; zum Beispiel Zeichnen von Linien, Marker und Textfolgen.

NDC:

Abkürzung für Normalisierte Koordinaten (normalized device coordinates).

Normalisierter Koordinatenraum (NDC-Raum):

Ein Normalisierter Koordinatenraum ist ein geräteunabhängiger virtueller Raum, durch welchen ein grafisches Nutzerprogramm die grafische Information an das SCP-GX übergibt. Im NDC-Raum liegen die Koordinatenwerte zwischen 0 und 32767. GDOS transformiert zwischen dem NDC-Raum und den Gerätekoordinaten (DC) eines speziellen Gerätes.

Normalisierte Koordinaten:

Normalisierte Koordinatensind Koordinaten des NDC-Raumes. Sämtliche Parameter in ptsin und ptsout werden bei Nutzung durch das Anwenderprogramm als Normalisierte Koordinaten angegeben.

Operationscode:

Der Operationscode wird als Bestandteil der Parameterliste an GDOS übergeben und zeigt an, welche grafische Operation gefordert ist.

Virtuelles Geräteinterface:

Das Virtuelle Geräteinterface ist ein Standardinterface zwischen geräteabhängigen und geräteunabhängigen Teilen der Grafik. VGI veranlaßt, daß alle Gerätedriver dem rufenden Programm gleich erscheinen. SCP-GX basiert auf dem Virtuellen Geräteinterface, und alle Gerätedriver, die für SCP-GX geschrieben sind, müssen an die VGI-Spezifikation angepaßt sein.

Arbeitsstation:

Eine Arbeitsstation ist eine grafische Geräteeinheit mit einer Darstellungsfläche und entweder keinem oder mehreren Eingabegeräten.

Arbeitsstations-Identifikationsnummer (ID):

Eine Arbeitsstations-ID ist eine logische Gerätenummer, die spezifiziert, welches grafische Gerät augenblicklich aktiv ist. Jeder Gerätedriver hat eine zugehörige Arbeitsstations-ID, welche in einer Zuweisungstabelle in der Datei ASSIGN.SYS spezifiziert wird.

Anlage 4 Gerätespezifische Informationen

1. Driver für das grafische Bildschirmsystem des A7100

1.1. Hardware

Das grafische Bildschirmsystem des A7100 (GBS) besteht aus den Teilen:
- KGS (Kontroller für grafisches Subsystem)
- ABG (Anschlußsteuerung für Bildschirm grafisch)
- Bildschirm K7229

Das GBS arbeitet in zwei Modi
- dem Alphanumerikmodus 25 Zeilen * 80 Spalten
- dem Grafikmodus 400 * 640 Bildpunkte

Im Grafikmodus existieren 2 Videoebenen, womit drei Graustufen realisiert werden können (Videoebene 1, Videoebene 2, Videoebenen 1 und 2).
Die Grafikfunktionen des GBS werden durch Firmware realisiert, die vor Absenden des ersten Grafikkommandos in den RAM des KGS geladen werden muß. Zum Laden der KGS-Firmware wird das Kommando L.CMD bereitgestellt.
Der Aufruf hat die Form

```
L GRAFx.FRM
```

GRAFx.FRM ist der Name der zu ladenden Firmware (x bezeichnet die Versionsnummer der Firmware).
Detaillierte Informationen zur GBS-Hardware sowie zu den grafischen Steueranweisungen des KGS sind in der Betriebsdokumentation (BAND 2) des A7100 enthalten.
Grafische Eingaben (LOCATOR, CHOICE, STRING und STROKE) sind möglich über die Geräte
- Tastatur K7637 oder
- grafisches Tablett K6405

Für den Betrieb des grafischen Tabletts ist mindestens Version 5 der KGS-Firmware notwendig.

1.2. Drivername und Geräteidentifikator

Name: KGSTBDRV.SYS

Der Identifikator für das Gerät wird in der Datei ASSIGN.SYS spezifiziert und sollte im Bereich 1-10 (CRT-Driver) liegen.

1.3. Arbeitsstationsbeschreibungstabelle

(Rückgabewerte beim OPEN WORKSTATION Ruf)

Tabelle 2: Arbeitsstationsbeschreibungstabelle beim Bildschirmsystem

Adresse		Inhalt
intout(1)	639	Größe der Darstellungsfläche in X-Richtung
intout(2)	399	Größe der Darstellungsfläche in Y-Richtung
intout(3)	1	Koordinatengenauigkeit
intout(4)	344	Breite eines Pixels in Mikrometern
intout(5)	344	Höhe eines Pixels in Mikrometern
intout(6)	1	Zeichenhöhen
intout(7)	5	Linientypen
intout(8)	7	Linienstärken
intout(9)	28	Markertypen
intout(10)	1	Markerhöhen
intout(11)	1	Textarten
intout(12)	0	Muster (Pattern)
intout(13)	8	Schraffuren
intout(14)	4	Farben
intout(15)	2	GDP's
intout(16)	1	GDP-Balken
intout(17)	2	GDP-Kreisbogen
intout(18)-(25)	-1	
intout(26)	3	Füllbereich
intout(27)	0	Polyline
intout(28)-(35)	-1	
intout(36)	1	Farbmöglichkeit (ja)
intout(37)	0	Textdrehung (nein)
intout(38)	1	Füllbereiche (ja)
intout(39)	1	Pixeloperationen (ja)
intout(40)	8	Größe der Farbtabelle
intout(41)	3	LOCATOR-Geräte
intout(42)	0	VALUATOR-Geräte
intout(43)	4	CHOICE-Geräte
intout(44)	3	STRING-Geräte
intout(45)	2	Arbeitsstationstyp
ptsout(2)	9	Zeichenhöhe in Pixel (min.)
ptsout(4)	9	Zeichenhöhe in Pixel (max.)
ptsout(5)	1	Linienstärke in Pixel (min.)
ptsout(7)	7	Linienstärke in Pixel (max.)
ptsout(10)	9	Markerhöhe in Pixel (min.)
ptsout(12)	9	Markerhöhe in Pixel (max.)

Die ptsout-Werte werden an den Nutzer transformiert in normalisierte Koordinaten (NDC) zurückgegeben.

1.4. Attribute für Grafikprimitiven

a. Linienarten:
1 Vollinie
2 Strichlinie
3 Punktlinie
4 Strich-Punkt-Linie
5 Strich-Punkt-Punkt-Linie

b. Markertypen:
1 Punkt
2 Kreuz waagerecht
3 Stern
4 Kreis (Kontur)
5 Kreuz (schräg)
6 Quadrat (Kontur)
7 Rhombus (Kontur)
8 Vollkreis
9 Vollquadrat
10 Vollrhombus
11 Pfeilspitze links
12 Pfeilspitze rechts
13 Pfeilspitze unten
14 Pfeilspitze oben
15-28 wie 1-14, aber stärker

c. Schraffuren:
1 senkrecht
2 waagerecht
3 45 Grad
4 135 Grad
5 senkrecht gekreuzt
6 schräg gekreuzt
7 Schachbrettmuster
8 verzerrtes Schachbrettmuster

d. Muster:
Es werden 5 fest programmierte und ein über die Funktion CELL ARRAY programmierbares Muster angeboten.
1 + -Zeichen
2 Quadrat, hell auf dunklem Hintergrund
3 Quadrat, dunkel auf hellem Hintergrund
4 * -Zeichen
5 Schachbrettmuster
6 über CELL ARRAY einstellbar
(Länge des Farbindexfeldes maximal 511)

e. Farben:
Das GBS ermöglicht, wie in Punkt 1 erwähnt, drei Graustufen, die der 8 Positionen umfassenden Farbtabelle, wie folgt zugeordnet werden.

Index	Videoebene	("Farbe")
0		(schwarz)
1	2	(rot)
2	1u. 2	(grün)
3	1	(blau)
4-7	2	(rot)

Die Zuordnung kann durch SPECIFY COLOR REPRESENTATION geändert werden. Dabei ist zu beachten, daß die Werte für die RGB-Intensitäten nur 0 oder 1000 sein können, wenn mehr als ein Wert ungleich 0 ist, wird immer Videoebene 2 (rot) ausgewählt.

1.5. ESCAPES

Gegenüber den in der Schnittstellenspezifikation enthaltenen Funktionen wurden folgende Änderungen bzw. Erweiterungen vorgenommen:

1.5.1. Funktion 17 - Hardcopy

Diese Funktion wurde um einen Parameter erweitert:

Eingabe

```
contrl(2) - 0
contrl(4) - 1
contrl(6) - Funktions-ID = 17
intin(1)  - Hardcopy-Typ: 00H - Druckeranschluß über ZVE,
                               Plotter-Modus
                          10H - Druckeranschluß über ZVE,
                               CRT-Modus
                          01H - Druckeranschluß über KGS,
                               Plotter-Modus
                          11H - Druckeranschluß über KGS,
                               CRT-Modus
```

Ausgabe

```
contrl(3) - 0
```

Bemerkung

Ist der Typ-Parameter nicht angegeben (contrl(4)=0), wird der Hardcopy-Typ 00H standardmäßig angenommen.

1.5.2. Funktion 20 - Setzen Splitgrenze

Eingabe

```
contrl(2) - 0
contrl(4) - 1
contrl(6) - Funktions-ID = 20
intin(1)  - Zeilennummer (1 - 25), ab der Alphanumerikinformationen
            sichtbar sein sollen.
```

Ausgabe

```
contrl(3) - 0
```

Beschreibung

Mit dieser Funktion kann der Nutzer auf dem A7100-Bildschirm die Grenze zwischen sichtbarer Grafik- und Alphanumerikinformation einstellen.

1.5.3. Funktion 21 - Spezifizieren Menüfeld

Eingabe

```
contrl(2) - 0
contrl(4) - 4
contrl(6) - Funktions-ID = 21
intin(1)  - Anzahl der Spalten des Menüfeldes (1 - 30)
intin(2)  - Anzahl der Zeilen des Menüfeldes (1 - 30)
intin(3)  - Länge eines Elementes des Menüfeldes in mm (10 - 90)
intin(4)  - Breite eines Elementes des Menüfeldes in mm (10 - 90)
```

Ausgabe

```
contrl(3) - 0
```

Beschreibung

Diese Funktion dient zur Beschreibung des Menüfeldes für CHOICE2-Eingaben vom Tablett. Die Standardeinstellung des Tabletts nach dem Einschalten ist 20/20/10/10.

1.5.4. Funktion 22 - Ausgabe eines KGS-Kommandos

Eingabe

```
contrl(2) - 0
contrl(4) - Länge des KGS-Kommandos (L512)
contrl(6) - Funktions-ID = 22
intin     - KGS-Kommando (1 Byte pro Wort)
```

Ausgabe

```
contrl(3) - 0
```

Beschreibung

Mittels dieser Funktion können Kommandos an den grafischen Bildschirmkontroller des A7100 übertragen werden, um spezielle nicht über das SCP-GX-Interface erreichbare Möglichkeiten des KGS nutzbar zu machen. Die KGS-Kommandos sind in der Betriebsdokumentation für den A7100 beschrieben.

Achtung: Die Nutzung dieser Funktion kann unter Umständen zu fehlerhaften Reaktionen des KGS-Treibers führen. Die KGS-Kommandos, die komplikationslos nutzbar sind, sind in der Kommandobeschreibung entsprechend gekennzeichnet.

1.6. Eingabegeräte (nur REQUEST-Eingaben)

Wie schon in Punkt 1.1. erwähnt, sind grafische Eingaben entweder über die Tastatur oder das grafische Tablett möglich. Das Standardeingabegerät (Gerätenummer 1) für die Eingabeklassen CHOICE und STRING ist immer die Tastatur. LOCATOR- und STROKE-Eingaben werden standardmäßig dem grafischen Tablett zugeordnet, wenn eins angeschlossen ist.

Punkt 1.6. beschreibt die Realisierung der grafischen Eingabeklassen LOCATOR, CHOICE, STRING und STROKE mittels Tastatur. Die Bedienung des grafischen Tabletts ist der Dokumentation zum grafischen Tablett K6405 zu entnehmen. Lediglich einige Besonderheiten, die sich aus der Zusammenarbeit mit dem grafischen Bildschirmsystem GBS ergeben, seien hier beschrieben:

- Das Tablett hat standardmäßig einen aktiven Bereich von 640 x 640 Punkten, das GBS einen Bildspeicher für 640 x 400 Punkte. Beim Befehl OPEN WORKSTATION wird der aktive Bereich des Tabletts an den Bildspeicher des GBS angepaßt. Das Tablett wird infolge dieser Anpassung in den RESET-Zustand versetzt und muß vom Nutzer durch Digitalisieren des Koordinatenursprungs wieder bereit gemacht werden.
- Zwei Felder des Standard-Menüfeldes erhielten eine spezielle Bedeutung zur Steuerung der Splitgrenze zwischen Alphanumerik- und Grafikbildschirm:
  POS - Verschieben der Splitgrenze nach oben
  ESTR - Verschieben der Splitgrenze nach unten
- Entsprechungen von Feldern des Standardmenüs des Tabletts und der Tastatur:

Tastatur	Tablett
CTRL/ENTER	Auslöseknopf der Lupe, wenn Lupe im aktiven Punktbereich
CR	CR
DEL	DEL
CTRL/X	CE
CTRL/9	BRK
CTRL/:	POS
CTRL/*	ESTR

a. LOCATOR-Eingabe:

Nach Absenden des Rufes INPUT LOCATOR erscheint auf dem Bildschirm ein Fadenkreuz. Dieses Fadenkreuz kann durch folgende Tasten des Zehnerfeldes der Tastatur in Verbindung mit CONTROL bewegt werden:

CTRL/= in die Bildmitte
CTRL/-> nach rechts
CTRL/<- nach links
CTRL/ nach oben
CTRL/ nach unten
CTRL/2 nach rechts oben
CTRL/- nach rechts unten
CTRL/1 nach links oben
CTRL/0 nach links unten

Der Abschluß einer LOCATOR-Eingabe erfolgt durch CTRL/9 (BREAK) ohne Rückgabe des LOCATOR-Wertes bzw. mit Rückgabe der Fadenkreuzposition durch CTRL/ENTER.

Desweiteren sind im LOCATOR-Modus zwei weitere Tasten im Zehnerfeld wirksam, mit denen die Splitgrenze zwischen Alphanumerik- und Grafikbildschirm verändert werden kann:

CTRL/: Verschieben der Splitgrenze nach oben
CTRL/* Verschieben der Splitgrenze nach unten

b. CHOICE-Eingabe:

Die CHOICE-Eingabe bezieht sich auf die Tasten PF1 bis PF12 der Tastatur.
Nach dem Betätigen einer dieser Tasten wird die Nummer der entsprechenden Funktionstaste zurückgegeben. CTRL/9 (BREAK) beendet die CHOICE-Eingabe, wenn die vom Nutzer gewählte Standardvorgabe der CHOICE-Nummer gültig werden soll.

c. STRING-Eingabe:

Über das STRING-Gerät können Alphazeichen an das Nutzerprogramm übermittelt werden. Die Eingabe wird erfolgreich beendet, wenn vom Nutzer CR eingegeben wurde.
Eingegebene Zeichen können vor Beenden der STRING-Eingabe durch DEL bzw. BS (Backspace) wieder gelöscht werden.
Das Löschen der gesamten Zeichenkette ist durch CTRL/X möglich.
Über die maximale STRING-Länge hinaus eingegebene Zeichen werden ignoriert.
Durch CTRL/9 (BREAK) kann die Texteingabe ohne Übergabe der bis dahin eingelesenen Zeichen abgebrochen werden. Im STRING-Eingabemodus sind die beiden Tasten CTRL/: und CTRL/* zur Steuerung des Splitmodus (s. a.) wirksam.

d. STROKE-Eingabe:

Über das STROKE-Gerät kann eine Folge von Punktkoordinaten (maximal 255) an das Nutzerprogramm übermittelt werden. Die Positionierung des Fadenkreuzes auf die gewünschten Eingabepunkte erfolgt in gleicher Weise wie bei a. (LOCATOR). Die Rückgabe der Koordinaten eines Punktes wird durch CTRL/ENTER ausgelöst.
Durch DEL kann der zuvor eingegebene Punkt wieder gelöscht werden, durch CTRL/X die gesamte Punktfolge. Die Darstellung der eingegebenen Punkte auf der Bildschirmfläche erfolgt entsprechend dem eingestellten Echo-Modus als Markerfolge, Linienzug oder Linienzug mit Markersymbolen.
Abgeschlossen wird eine STROKE-Eingabe durch CR. Durch CTRL/9 (BREAK) kann die STROKE-Eingabe ohne Übergabe der bis dahin eingegebenen Koordinaten abgebrochen werden.
Über die maximale STROKE-Länge hinaus eingegebene Punkte werden ignoriert.

1.7. GDP

Verfügbar sind die GDP's Balken und Kreisbogen. Beim Kreisbogen ist folgendes zu beachten:
- Ist contrl[4] = 0 (kein Winkel angegeben), wird der Kreisbogen aus Mittelpunkt, Anfangs- und Endpunkt bestimmt.
- Ist contrl[4] = 2, wird der Kreisbogen aus Mittelpunkt, Radius, Anfangs- und Endwinkel bestimmt.
- Ein Vollkreis wird generiert bei Angabe von 0 für den Anfangswinkel und 3600 für den Endwinkel (contrl[4] = 2)

1.8. Zusammenstellung der verwendbaren GIOS-Funktionen

1	OPEN WORKSTATION
2	CLOSE WORKSTATION
3	CLEAR WORKSTATION
4	UPDATE WORKSTATION
5	ESCAPE
6	POLYLINE
7	POLYMARKER
8	TEXT
9	FILL AREA(maximal 128 Punkte)
10	CELL ARRAY
11	GDP (Balken, Kreisbogen)
12	SET CHARACTER HIGH
14	SET COLOR REPRESENTATION
15	SET POLYLINE LINETYP
16	SET POLYLINE LINEWIDTH
17	SET POLYLINE COLOR INDEX
18	SET POLYMARKER TYP
20	SET POLYMARKERCOLOR INDEX
22	SET TEXT COLOR INDEX
23	SET INTERIOR FILL STYLE
24	SET STYLE INDEX
25	SET FILL COLOR INDEX
26	INQUIRE COLOR REPRESENTATION
28	INPUT LOCATOR
30	INPUT CHOICE
31	INPUT STRING
32	SET WRITING MODE (Ersetzen, ODER-Verknüpfung, Löschen)
33	SET INPUT MODE (Request)
34	INPUT STROKE

2. Driver für den grafischen Drucker K6313 des A7100

2.1. Hardware

Der Hardcopy-Drucker robotron K6313 ist ein mittelschneller, serieller Matrixdrucker mit Nadeldrucksystem.
Er wird über die Parallelschnittstelle CENTRONICS an den A7100 angeschlossen.
Die Druckgeschwindigkeit beträgt 100 Zeichen/Sekunde. Es können ein Original und zwei Durchschläge bei einer maximalen Druckbreite von 216 mm hergestellt werden. Die Druckrichtung ist vorwärts und rückwärts mit Druckwegoptimierung.

2.2. Drivername und Geräteidentifikator

a. Name: K6313HR8.SYS bzw. K6313LR8.SYS

Der Identifikator für das Gerät wird in der Datei ASSIGN.SYS spezifiziert und sollte im Bereich 21-30 (Drucker-Driver) liegen.

b. Unterschied:

Der K6313HR8.SYS arbeitet langsamer (druckt doppelt) als der K6313LR8.SYS, erzeugt jedoch ein feineres Bildraster.
Der K6313LR8.SYS sollte mehr zum Testen eines Grafikbildes benutzt werden, um schnell einen Überblick zu erhalten. Das eigentliche Grafikbild kann man exakter mit dem K6313HR8.SYS erhalten.
Im Funktionsumfang sind beide Driver gleich.

2.3. Arbeitsstationsbeschreibungstabelle

(Rückgabewerte beim OPEN WORKSTATION Ruf)

Tabelle 3: Arbeitsstationsbeschreibungstabelle für Drucker

Adresse	Inhalt a)	Inhalt b)	Bedeutung
intout(1)	959	479	Größe der Darstellungsfläche in X-Richtung
intout(2)	1367	671	Größe der Darstellungsfläche in Y-Richtung
intout(3)	0	0	Koordinatengenauigkeit
intout(4)	212	422	Breite eines Pixels in Mikrom.
intout(5)	176	352	Höhe eines Pixels in Mikrom.
intout(6)	12	12	Zeichenhöhen
intout(7)	6	6	Linientypen
intout(8)	1	1	Linienstärken
intout(9)	5	5	Markertypen
intout(10)	12	12	Markerhöhen
intout(11)	1	1	Textarten
intout(12)	0	0	Muster (Pattern)
intout(13)	6	6	Schraffuren
intout(14)	2	2	Farben
intout(15)	1	1	GDP's
intout(16)	1	1	Balken (nicht implementiert)
intout(17)-(25)	-1	-1	
intout(26)	3	3	Füllbereich
intout(27)-(35)	-1	-1	
intout(36)	0	0	Farbmöglichkeit (nein)
intout(37)	1	1	Textdrehung (ja)
intout(38)	1	1	Füllbereiche (ja)
intout(39)	0	0	Pixeloperationen (nein)
intout(40)	2	2	Größe der Farbtabelle
intout(41)-(44)	0	0	Eingabegeräte
intout(45)	0	0	Arbeitsstationstyp
ptsout(2)	8	8	Zeichengröße in Pixel (min.)
ptsout(4)	96	96	Zeichengröße in Pixel (max.)
ptsout(5)	1	1	Linienstärke in Pixel (min.)
ptsout(7)	1	1	Linienstärke in Pixel (max.)
ptsout(10)	8	8	Markergröße in Pixel (min.)
ptsout(12)	96	96	Markergröße in Pixel (max.)

Die ptsout-Werte werden an den Nutzer transformiert in normalisierte Koordinaten (NDC) zurückgegeben.

Bemerkung: a) K6313HR8.SYS
b) K6313LR8.SYS

2.4. Attribute für Grafikprimitiven

a. Linienarten: 1 Vollinie
2 Strichlinie (kurz)
3 Punktlinie
4 Strich-Punkt-Linie

	5	Strichlinie (lang)
	6	Strich-Punkt-Punkt-Linie
b. Markertypen:	1	Punkt
	2	Kreuz (waagerecht)
	3	Stern
	4	Quadrat
	5	Kreuz (schräg)
c. Schraffuren:	1	senkrecht
	2	waagerecht
	3	+45 Grad
	4	-45 Grad (+135 Grad)
	5	voll ausgefüllt
	6	voll ausgefüllt
d. Farben:	0	Weiß (kein Druck)
	1	Schwarz
e. Schreibmodus:	1	Ersetzen
	2	Überschreiben (ODER-Verknüpfung)
	3	Komplement
	4	Löschen
f. Innere Ausfüllart:	0	hohl
	1	voll
	2	gemustert
	3	schraffiert

2.5. Zusammenstellung der verwendbaren GIOS-Funktionen

1	OPEN WORKSTATION
2	CLOSE WORKSTATION
3	CLEAR WORKSTATION
4	UPDATE WORKSTATION
6	POLYLINE
7	POLYMARKER
8	TEXT
9	FILL AREA
10	CELL ARRAY
11	GDP (nicht implementiert)
12	SET CHARACTER HEIGHT
13	SET CHARACTER UP VECTOR (um jeweils 45 Grad)
15	SET POLYLINE LINETYP
17	SET POLYLINE COLOR INDEX
18	SET POLYMARKER TYPE
19	SET POLYMARKER SCALE
20	SET POLYMARKER COLOR INDEX
22	SET TEXT COLOR INDEX
23	SET FILL INTERIOR STYLE
24	SET FILL STYLE INDEX
25	SET FILL COLOR INDEX
32	SET WRITING MODE

3. Grafischer Driver für den Plotter K6418 am A7100

3.1. Hardware

Der Plotter wird am A7100 als grafisches Ausgabegerät benutzt und dazu entweder an das IFSS oder an das IFSP der ASP K8071 angeschlossen.
Die nutzbare Schreibfläche mißt in der X-Achse 370mm, in der Y-Achse 270mm. Die kleinste adressierbare Schrittgröße beträgt für beide Achsen 0,1mm. Die Zeichengeschwindigkeit in Achsrichtung ist in Stufen einstellbar (6/12/24 cm/s). Der Plotter verfügt über einen integrierten 8-Bit- Mikroprozessor (U880) und entsprechende Firmware, so daß eine Reihe Plotkommandos einschließlich der Generierung von ASCII-Zeichen und von kyrillischen Großbuchstaben im Gerät selbst realisiert werden. Die Ansteuerung des Plotters vom grafischen Driver erfolgt über 16 Kommandos für absolutes und relatives Plotten und zur Textdarstellung, wobei die Kommandos und ihre Parameter als ASCII-Zeichen übertragen werden.
Nähere Informationen zum K6418 sind der entsprechenden Gerätedokumentation zu entnehmen.

3.2. Drivername und Geräteidentifikator

Der Driver liegt in 4 Varianten vor:

K6418S.SYS:	Driver für K6418 am IFSS der ASP ohne Unterstützung von Darstellungs-Primitiven (generalized drawing primitives=GDP) entsprechend SCP-GX.
K6418SG.SYS:	Driver für K6418 am IFSS der ASP mit Unterstützung von GDP.
K6418P.SYS:	Driver für K6418 am IFSP der ASP ohne Unterstützung von GDP.
K6418PG.SYS:	Driver für K6418 am IFSP der ASP mit Unterstützung von GDP.

Die Unterstützung von GDP ist auf das Zeichnen der Konturen der GDP beschränkt, d.h. der "FILL INTERIOR STYLE" ist stets 0 = hollow (leer).
In allen Driver-Varianten werden dieselben GDP-Aufrufe verarbeitet, in den Varianten K6418S.SYS und K6418P.SYS wird jedoch bei OPEN WORKSTATION zurückgemeldet, daß keine GDP unterstützt werden. Dadurch können entsprechende Anwenderprogramme veranlaßt werden, im Bedarfsfall Flächen mit geeigneter Füllung in eigener Regie zu erzeugen.
Der Geräteidentifikator für den Plotter wird in der Datei ASSIGN.SYS dem grafischen Driver zugeordnet und sollte im Bereich 11...20 liegen.

3.3. Arbeitsstations-Beschreibungstabelle

Beim SCP-GX-Ruf 1 (OPEN WORKSTATION) werden folgende gerätespezifische Informationen an das rufende Applikationsprogramm zurückgegeben.

Tabelle 4: Arbeitsstationsbeschreibungstabelle für PlotterK6418

Adresse	Inhalt	Bedeutung
intout(1)	3699	max. adressierbare Weite (X-Achse) des Plotters in Rasterschritten
intout(2)	2699	max. adressierbare Höhe (Y-Achse) des Plotters in Rasterschritten
intout(3)	0	Flag für Gerätekoordinatengenauigkeit
intout(4)	100	Weite eines Plotterschrittes in Mikrom.
intout(5)	100	Höhe eines Plotterschrittes in Mikrom.
intout(6)	0	Anzahl von Zeichenhöhen (kontinuierlich einstellbar)
intout(7)	6	Anzahl von Linientypen
intout(8)	1	Anzahl von Linienstärken
intout(9)	5	Anzahl von Markertypen
intout(10)	0	Anzahl von Markerhöhen (kontinuierlich einstellbar)
intout(11)	2	Anzahl von Textarten (fonts)
intout(12)	0	Anzahl von Mustern (Pattern)
intout(13)	0	Anzahl von Schraffuren (hatch styles)
intout(14)	2	Anzahl der vordefinierten Farben (monochromatisches Gerät)
	a) b)	
intout(15)	5 0	Anzahl der GDP
intout(16)	1 -1	Rechteck-GDP (BAR)
intout(17)	2 -1	Kreisbogen-GDP (ARC)
intout(18)	3 -1	Kreissegment-GDP (PIE SLICE)
intout(19)	4 -1	Kreis-GDP (CIRCLE)
intout(20)	5 -1	Text-GDP (PRINT GRAPHIC CHARACTERS)
intout(21)-(25)	-1	(keine weiteren GDP definiert)
intout(26)	0 -1	Attribut-Satz für Rechteck-GDP (entspr. polyline)
intout(27)	0 -1	Attribut-Satz für Kreisbogen-GDP (entspr. polyline)
intout(28)	0 -1	Attribut-Satz für Kreissegment-GDP (entspr. polyline)
intout(29)	0 -1	Attribut-Satz für Kreis-GDP (entspr. polyline)
intout(30)	2 -1	Attribut-Satz für Text-GDP (entspr. text)
intout(31)-(35)	-1	(keine weiteren GDP-Attribute)
intout(36)	0	Flag für Farbfähigkeit (nein)
intout(37)	1	Flag für Textdrehung (ja)
intout(38)	0	Flag für Flächendrehung (nein)
intout(39)	0	Flag für Pixeloperationen (nein)
intout(40)	2	Anzahl verfügbarer Farben (monochrom.)
intout(41)	0	Anzahl der LOCATOR-Geräte
intout(42)	0	Anzahl der VALUATOR-Geräte
intout(43)	0	Anzahl der CHOICE-Geräte
intout(44)	0	Anzahl der STRING-Geräte
intout(45)	0	Arbeitsstationstyp (nur output)

Tabelle 4: Fortsetzung

Adresse	Inhalt	Bedeutung
ptsout(1)	0	
ptsout(2)	36	min. Zeichenhöhe in Geräteeinheiten (Plotterschritten)
ptsout(3)	0	
ptsout(4)	600	max. Zeichenhöhe in Geräteeinheiten
ptsout(5)	1	min. Linienbreite in Geräteeinheiten
ptsout(6)	0	
ptsout(7)	1	max. Linienbreite in Geräteeinheiten
ptsout(8)	0	
ptsout(9)	0	
ptsout(10)	24	min. Markerhöhe in Geräteeinheiten
ptsout(11)	0	
ptsout(12)	600	max. Markerhöhe in Geräteeinheiten

Die ptsout-Werte werden an den Nutzer transformiert in normalisierte Koordinaten (NDC) zurückgegeben.

Anmerkungen zu den Werten in intout (1), (2):
Die Werte gelten für den Fall, daß die gesamte Darstellungsfläche des Plotters K6418 genutzt wird. Wird dagegen mit einem Viewport, d. h. mit einem rechteckigen, achsenparallelen Teilbereich der Darstellungsfläche (s. Pkt. 3.6.5. in Anlage 4) gearbeitet, werden die entsprechenden Werte für das gewählte Viewport angegeben.

Anmerkung zu den Werten in ptsout(1), (4), (10), (12):
Die Höhe von Textzeichen und damit auch von Markersymbolen kann als Vielfaches eines Basiszeichenrasters (6 Punkte in der Höhe) eingestellt werden. Die o. g. Werte für minimale und maximale Zeichen- bzw. Markerhöhe sind zweckmäßige aber keine verbindlichen Festlegungen.

Bemerkung: a) für K6418SG.SYS und K6418PG.SYS
b) für K6418S.SYS und K6418P.SYS

3.4. Attribute für die Darstellungselemente

a. Linienarten:

Mit dem SCP-GX-Ruf 15 (SET POLYLINE LINETYP) wird durch Angabe einer Nummer die Linienart für nachfolgende POLYLINE-Rufe festgelegt:

1 Vollstrich
2 Strichlinie im Teilungsverhältnis (Strich:Lücke) 1:1
3 Punktlinie
4 Strich-Punkt-Linie
5 Strich-Punkt-Punkt-Linie
6 Strichlinie (Langstrich-Kurzstrich-Kurzstrich)

b. Markertypen:

Mit dem SCP-GX-Ruf 18 (SET POLYMARKER TYPE) wird durch Angabe einer Nummer der Markertyp für nachfolgende POLYMARKER-Rufe

festgelegt:

1	. (Punkt)
2	+ (Plus-Zeichen
3	* (Stern)
4	o (Kleinbuchstabe o)
5	x (Kleinbuchstabe x)

c. Schriftarten:

Unabhängig von der Größe der Schrift können über den SCP-GX-Ruf 21 (SET TEXT FONT) 2 Schriftarten eingestellt werden:

1	gerade Schrift (Neigung 0 Grad)
2	schräge Schrift (Neigung 18 Grad)

d. Farben:

Der Plotter K6418 ist als monochromatisches Gerät zu betrachten, wenngleich Faserstifte unterschiedlicher Farbe manuell (!) gewechselt werden können. Deshalb werden nur 2 Farbindices unterschieden:

0	Papier, d. h. keine Farbe
1	Tinte, d. h. Farbe des eingesetzten Schreibstiftes

3.5. ESCAPES

Von den in der SCP-GX-Beschreibung genannten ESCAPE-Funktionen wird nur die Funktion 1 (INQUIRE ADDRESSABLE CHARACTER CELLS) realisiert.
Darüber hinaus wurden zusätzliche ESCAPE-Funktionen definiert und im Driver realisiert, um spezielle Leistungen des Plotters K6418 auch über die SCP-GX-Schnittstelle verfügbar zu machen. Diese Funktionen werden im folgenden beschrieben.

3.5.1. SET CHARACTER SLANT

Neigungswinkel für Textzeichen einstellen

Eingabe: contrl(1)=5 (Operationscode für ESCAPE)
contrl(4)=1
contrl(6)=51
intin(1) =slant-Nr.

Ausgabe: contrl(3)=0

Beschreibung: Mit dieser ESCAPE-Funktion wird über eine slant-Nr. der Neigungswinkel von Textzeichen, gemessen von der Senkrechten auf der Basislinie des Textes, eingestellt. Zwischen slant-Nr. und Neigungswinkel besteht folgender Zusammenhang:

slant-Nr.	0	&1	&2	&3	&4	&5	&6	&7	&8
Neigungs-winkel in Grad	0	&9	&18	&26	&34	&40	&45	&49	&53

Bem.: Das Zeichen & entspricht +/-.

Für slant-Nr. >8 wird vom Driver 8, für slant-Nr. <8 wird -8 angenommen.

3.5.2. SET CHARACTER PROPORTION

Höhe und Breite für Textzeichen einstellen

Eingabe: contrl(1)=5
contrl(2)=1
contrl(6)=52
ptsin(1) =Zeichenbreite in NDC
ptsin(2) =Zeichenhöhe in NDC

Ausgabe: contrl(3)=0

Beschreibung: Im Unterschied zur SCP-GX-Funktion 12 (SET CHARACTER HEIGHT), bei der die Zeichengröße mit gleichbleibendem Seitenverhältnis (Breite zu Höhe) variiert werden kann, gestattet diese ESCAPE-Funktion eine beliebige Einstellung von Breite und Höhe für Textzeichen, d.h. die Auswahl einer beliebigen Zeichenproportion für nachfolgende Textausgaben.

3.5.3. MOVE PLOT POSITION

Verändern der aktuellen Schreibposition im Zusammenhang mit Textausgaben

Eingabe: contrl(1)=5
contrl(4)=2
contrl(6)=53
intin(1) =Anzahl von Leerzeichen n } positive und
intin(2) =Anzahl von Zeilen m } negative Werte
} zugelassen

Ausgabe: contrl(3)=0

Beschreibung: Entsprechend der aktuellen Zeichengröße und Schreibrichtung wird der Schreibwagen um n Zeichenpositionen nach rechts/links um m Zeilen nach unten/oben bewegt.

3.5.4. SET CHARACTER SET

Einstellen Zeichensatz

Eingabe: contrl(1)=5
contrl(4)=1
contrl(6)=54
intin(1) =Zeichensatz-Nr.

Ausgabe: contrl(3)=0

Beschreibung: Entsprechend der angegebenen Zeichensatz-Nr. wird der Zeichensatz für folgende Textausgaben ausgewählt:
Zeichensatz-Nr. 1: latainischer Zeichensatz
Zeichensatz-Nr. 2: kyrillischer Zeichensatz
(den Code-Positionen 60H bis 7EH werden kyrillische Großbuchstaben zugeordnet)

3.5.5. SET TICK LENGTH

Einstellen der Länge von Skalierungsstrichen auf den Koordinatenachsen

Eingabe: contrl(1)=5
contrl(4)=2
contrl(6)=55
intin(1) =positive Skalierungsstrichlänge in DC
intin(2) =negative Skalierungsstrichlänge in DC

Ausgabe: contrl(3)=0

Beschreibung: Mit dieser ESCAPE-Funktion wird die positive (für Skalenstriche auf der X-Achse oberhalb der X-Achse, für Skalenstriche auf der Y-Achse rechts von der Y-Achse) und negative (unterhalb der X-Achse bzw. links von der Y-Achse) Länge nachfolgend zu zeichnender Skalenstriche eingestellt.
Es ist zu beachten, daß diese Längenangaben in DC-Einheiten, d.h. in Plotterschritten von 0,1mm, übergeben werden müssen.

3.5.6. DRAW X-AXIS-TICK

Zeichnen eines Skalenstriches auf der X-Achse

Eingabe: contrl(1)=5
contrl(6)=56

Ausgabe: contrl(3)=0

Beschreibung: An der aktuellen Schreibwagenposition (auf der X-Achse) wird ein Skalenstrich entsprechend der voreingestellten Länge gezeichnet.

3.5.7. DRAW Y-AXIS-TICK

Zeichnen eines Skalenstriches auf der Y-Achse

Eingabe: contrl(1)=5
contrl(6)=57

Ausgabe: contrl(3)=0

Beschreibung: An der aktuellen Schreibwagenposition (auf der Y-Achse) wird ein Skalenstrich entsprechend der voreingestellten Länge gezeichnet.

3.5.8. SELECT VELOCITY

Einstellen der Zeichengeschwindigkeit

Eingabe: contrl(1)=5
contrl(6)=58
intin(1) =Geschwindigkeits-Nr.

Ausgabe: contrl(3)=0

Beschreibung: Entsprechend der Geschwindigkeits-Nr. wird die Zeichengeschwindigkeit in Achsrichtung eingestellt. Dabei gilt:

Geschwindigkeits-Nr.	0	1	2
Geschwindigkeit in mm/s	60	120	240

3.5.9. ROTATE COORDINATE SYSTEM

Drehen des Koordinatensystems

Eingabe: contrl(1)=5
contrl(6)=59
intin(1) =Drehwinkel in Grad

Ausgabe: contrl(3)=0

Beschreibung: Das Koordinatensystem wird um den angegebenen Winkel gedreht. Drehpunkt ist die aktuelle Schreibwagenposition.
Die Drehung ist wirksam beim Zeichnen mit relativen Koordinaten (s. Pkt. 5.10. und 5.11.)

3.5.10. PLOT RELATIV (POLYLINE)

Zeichnen von Polyline mit relativen Koordinaten

Eingabe: contrl(1)=5
contrl(4)=n (n=Anzahl der relativen Koordinatenpaare)
contrl(6)=60
intin(1) =relat. X-Koord. d. 1. Punktes in NDC
intin(2) =relat. Y-Koord. d. 1. Punktes in NDC
.
.
.
intin(2n-1)=rel. X-Koord. d. n-ten Punktes in NDC
intin(2n)=relat. Y-Koord. d. n-ten Punktes in NDC

Ausgabe: contrl(3)=0

Beschreibung: Ausgehend von der aktuellen Schreibwagenposition wird eine Polyline entsprechend der angegebenen relativen Koordinaten gezeichnet, wobei die Koordinaten positive oder negative Integerzahlen (-32766...+32767) sein können.

3.5.11. PLOT RELATIV (ARC)

Zeichnen eines Kreisbogens mit relativen Mittelpunktskoordinaten

```
Eingabe: contrl(1)=5
         contrl(4)=5
         contrl(6)=61
         intin(1) =Startwinkel in Zehntelgrad
         intin(2) =Endwinkel in Zehntelgrad
         intin(3) =Drehrichtung
         intin(4) =relat. X-Koord. des Kreismittelp. in NDC
         intin(5) =relat. Y-Koord. des Kreismittelp. in NDC

Ausgabe: contrl(3)=0
```

Beschreibung: Mit dieser Funktion wird ab der aktuellen Schreibwagenposition ein Kreisbogen um den relativ zum Startpunkt festgelegten Mittelpunkt gezeichnet. Start- und Endwinkel werden entgegen dem Uhrzeigersinn ab der positiven X-Achse (Winkel 0) gemessen. Für Drehrichtung=0 wird der Bogen vom Start- zum Endpunkt entgegen dem Uhrzeigersinn, für Drehrichtung ungleich 0 wird der Bogen vom Start- zum Endpunkt im Uhrzeigersinn gezeichnet.

3.5.12. SET WINDOW

Einstellen der Grenzen eines Fensters auf der Darstellungsfläche

```
Eingabe: contrl(1)=5
         contrl(2)=2
         contrl(6)=62
         ptsin(1) =X-Koord. der linken unteren Ecke in NDC
         ptsin(2) =Y-Koord. der linken unteren Ecke in NDC
         ptsin(3) =X-Koord. der rechten oberen Ecke in NDC
         ptsin(4) =Y-Koord. der rechten oberen Ecke in NDC

Ausgabe: contrl(3)=0
```

Beschreibung: Durch Angabe der Koordinaten für die linke untere und die rechte obere Ecke wird ein rechteckiges Fenster auf der Darstellungsfläche des Plotters definiert. Nur innerhalb dieses Fensters liegende Elemente eines Bildes werden gezeichnet.

3.5.13. UNIVERSAL ESCAPE-FUNCTION

Universelle ESCAPE-Funktion zur direkten Ausgabe von Plotterbefehlen

```
Eingabe: contrl(1)=5
         contrl(4)=Anzahl der ASCII-Zeichen in intin
         contrl(6)=99
         intin(1) =1. und 2. Zeichen                    \
         intin(2) =3. und 4. Zeichen                    |  des Strings
             .                                           \ von
             .                                           / Plotter-
             .                                           |  befehlen
         intin(n) =(2n-1)-tes (und (2n)-tes) Zeichen    /

Ausgabe: contrl(3)=0
```

Beschreibung: Die im intin-Bereich (gepackt) stehende Kette von ASCII-Zeichen, die gültige Plotterbefehle darstellen, wird direkt an den Plotter ausgegeben. Die Anzahl der auszugebenden Zeichen ist in contrl(4) bereitzustellen.

3.6. Hinweise zur Arbeit mit SCP-GX-Funktionen für Ausgaben zum Plotter K6418

3.6.1. Koordinatenkorrektur

Der Plotter K6418 hat in X-Richtung 3700 und in Y-Richtung 2700 adressierbare Punkte.
Das bedeutet, daß eine in NDC-Einheiten gegebene Strecke auf dem Plotter in X-Richtung größer abgebildet wird als in Y-Richtung.
Die Ursache dafür ist die SCP-GX-Konvention, daß die NDC-Koordinaten 0-32767 voll auf jede Achse des Ausgabegerätes abgebildet werden.
Soll also auf dem Ausgabegerät eine unverzerrte, d.h. in X- und Y-Richtung äquidistante Darstellung erfolgen, muß im Anwenderprogramm eine Koordinatenkorrektur für eine Achse vorgesehen werden.

Bei Korrektur der Y-Koordinaten ist der Korrekturfaktor

YKor = 3700/2700 = 1,37

und analog bei Korrektur der X-Koordinaten

XKor = 2700/3700 = 0,73

mit den entsprechenden Koordinatenwerten zu multiplizieren.

3.6.2. Textdarstellung

Textzeichen werden von der Plotter-Firmware erzeugt, wobei von einem Basisraster 5 (Breite) mal 8 (Höhe) ausgegangen wird. Innerhalb dieses Rasters gelten folgende Proportionen:

	Breite	Höhe	Zeichenabstand
Großbuchstaben	4	6	1
Kleinbuchstaben	3	4(6)	2
Ziffern	3	6	2
Unterlänge	-	2	-
Zeilenabstand	-	4	-
Space/Backspace	5/-5	-	-

Mit der SCP-GX-Funktion 12 (SET CHARACTER HEIGHT) bzw. 5 (ESCAPE), Funktion 52 (SET CHARACTER PROPORTION) werden die Höhen von Großbuchstaben festgelegt. Aus dem Basisraster ergeben sich die anderen Abmessungen als Vielfache der Basisbreite bzw. Basishöhe.

Mit der SCP-GX-Funktion 13 (SET CHARACTER UP VECTOR) wird die Richtung der Basislinie für nachfolgende Textausgaben festgelegt. Dabei wird in intin(1) der Winkel zwischen Basislinie und positiver X-Achse (gemessen entgegen dem Uhrzeigersinn) übergeben. Der

Driver nutzt diesen Winkel für einen plottereigenen Rotate-Befehl.
Über die ESCAPE-Funktion 99 kann mit dem Plotterbefehl "DR-1,0;" Spiegelschrift eingestellt und mit dem Plotterbefehl "DR 1,0;" auf Normalschrift zurückgestellt werden.
Bei Ausgabe von Text über die SCP-GX-Funktion 8 bzw. 11 wird eine Umschaltung in den kyrillischen Zeichensatz veranlaßt, wenn mindestens 1 Zeichen im Ausgabestring ein gesetztes 8. Bit (alternativer Zeichensatz) hat. Eine gemischte Ausgabe von lateinischen und kyrillischen Zeichen innerhalb derselben String ist deshalb nicht möglich. Kyrillische Kleinbuchstaben im Ausgabestring werden in kyrillische Großbuchstaben umgesetzt.

3.6.3. GDP für Kreisbogen bzw. Kreissegment

Die Aufrufparameter für diese GDP (siehe Anlage 2, Pkt. 11) sind redundant. Folgende Untermengen sind beim Aufruf durch das AP für die Ausgabe zum Plotter ausreichend:

a) contrl(2)=3, d. h. ohne Angabe des Radius in ptsin(7)
   contrl(4) ungleich 2, d. h. ohne Winkelangabe in intin
b) contrl(2)=2, d. h. nur Mittelpunkt und Anfangspunkt in
   ptsin(1)...ptsin(4)
   contrl(4)=2, d. h. Anfangs- und Endwinkel in intin(1)
   und intin(2)
c) contrl(2)=4, nur Mittelpunkt in ptsin(1/2) und
   Radius in ptsin(7)
   contrl(4)=2, d. h. Anfangs- und Endwinkel in intin(1)
   und intin(2)

3.6.4. Plotten mit relativen Koordinaten

Abweichend von den SCP-GX-Konventionen werden beim Plotten mit relativen Koordinaten (siehe Pkt. 3.5.10. und 3.5.11.) diese Koordinaten im intin-Bereich übergeben, da im ptsin-Bereich nur positive Werte (0...32767) zugelassen sind. Die für ptsin-Werte übliche Transformationen zwischen NDC und DC wird für die relativen Plot-Koordinaten im intin-Bereich im Driver durchgeführt, so daß diese Koordinaten wie beim absoluten Plotten als NDC-Werte anzugeben sind.

3.6.5. Nutzung eines Viewports

Normalerweise wird der gesamte NDC-Koordinatenraum auf die physisch vorhandene Darstellungsfläche des Ausgabegerätes abgebildet.
Mitunter - vor allem bei Ausgaben auf Plotter - ist es jedoch notwendig, für die Darstellung eines Bildes nur einen Teil der vorhandenen Darstellungsfläche zu nutzen. Im Unterschied zu der Arbeit mit einem Fenster (s. Pkt. 3.5.12.), bei der nur ein im Rahmen des Fensters liegender Bildausschnitt in ursprünglicher Größe dargestellt wird, bewirkt das Einstellen eines Viewports eine Abbildung des gesamten NDC-Koordinatenraumes auf die Fläche des Viewport, d. h. das gesamte Bild wird im Viewport verkleinert dargestellt.
Das Viewport ist ein rechteckiger, achsenparalleler Bereich, dessen DC-Koordinaten bei OPEN WORKSTATION wie folgt angegeben werden:

a) linke untere Ecke des Viewport stimmt mit linker unterer Ecke der physischen Darstellungsfläche überein:
   contrl(4)=12
   intin(11)=x-Koordinate der rechten oberen Ecke des Viewport in DC
   intin(12)=y-Koordinate der rechten oberen Ecke des Viewport in DC
b) Viewport hat beliebige Lage auf der physischen Darstellungsfläche:
   contrl(4)=14
   intin(11),(12) siehe Fall a)
   intin(13)=x-Koordinate der linken unteren Ecke des Viewport in DC
   intin(14)=y-Koordinate der linken unteren Ecke des Viewport in DC

3.7. Zusammenstellung der verwendbaren SCP-GX-Funktionen

```
 1      OPEN WORKSTATION
 2      CLOSE WORKSTATION
 3      CLEAR WORKSTATION
 4      UPDATE WORKSTATION
 5      ESCAPE (MIT DEN FUNKTIONEN 1, 51...62, 99)
 6      POLYLINE
 7      POLYMARKER
 8      TEXT
 9      FILLED AREA (NUR UMRANDUNG)
10      CELL ARRAY (NUR UMRANDUNG)
11      GDP (BALKEN, KREISBOGEN, KREISSEGMENT, KREIS, TEXT)
12      SET CHARACTER HEIGHT
13      SET CHARACTER UP VECTOR
14      SET COLOR REPRESENTATION *)
15      SET POLYLINE LINETYPE
17      SET POLYLINE COLOR INDEX
18      SET POLYMARKER TYPE
19      SET POLYMARKER SCALE
20      SET POLAMARKER COLOR INDEX
21      SET TEXT FONT
22      SET TEXT COLOR INDEX
23      SET FILL INTERIOR STYLE
25      SET FILL COLOR INDEX
26      INQUIRE COLOR REPRESENTATION
```

*) ohne spezielle Wirkung

4. Grafischer Driver für den Plotter K6411 am A7100

4.1. Hardware

Der Plotter K6411 wird primär als grafisches Ausgabegerät verwendet, ermöglicht aber auch eine grafische Eingabe als Locator-Gerät (Digitalisiermodus). Er wird beim A7100 an das IFSS der ASP K8071 angeschlossen. Der Plotter gestattet Zeichen- und Beschriftungsarbeiten bis zu einem Papierformat A2 (420mmx594mm). In der x-Achse sind 24160 und in der y-Achse 17600 Positionen bei einer Schrittweite von 0,025mm ansteuerbar.
Der Plotter kann ohne Unterbrechung der Programmabarbeitung bis zu 8 verschiedene Schreibwerkzeuge benutzen (automatischer Wechsel der Scheibwerkzeuge). Durch die integrierte Firmware wird eine spezielle Plottersprache unterstützt, die eine Ansteuerung

des Plotters durch eine Kommandofolge im ASCII-Format erlaubt. Die Firnware gewährleistet auch die Darstellung von Schriftzeichen unterschiedlicher Zeichensätze (einschließlich kyrillischer Klein- und Großbuchstaben).
Nähere Informationen zum K6411 sind der entsprechenden Gerätedokumentation zu entnehmen.

4.2. Drivername und Geräteidentifikator

Der Driver liegt in 2 Varianten vor:

K6411S.SYS — Driver für K6411 am IFSS der ASP ohne Unterstützung von Darstellungs-Primitiven (GDP) entsprechend SCP-GX

K6411SG.SYS — Driver für K6411 am IFSS der ASP mit Unterstützung von GDP.

Die Unterstützung von GDP ist auf das Zeichnen der Konturen der GDP beschränkt, d.h. der "FILL INTERIOR STYLE" ist stets 0 = hollow (leer). In allen Driver-Varianten werden dieselben GDP-Aufrufe verarbeitet, in der Variante K6411S.SYS wird jedoch bei OPEN WORKSTATION zurückgemeldet, daß keine GDP unterstützt werden. Dadurch können entsprechende Anwenderprogramme veranlaßt werden, im Bedarfsfall Flächen mit geeigneter Füllung bzw. spezielle Textfonds in eigener Regie zu erzeugen.
Der Geräteidentifikator für den Plotter wird in der Datei ASSIGN.SYS dem grafischen Driver zugeordnet und sollte im Bereich 11..20 liegen.

4.3. Arbeitsstationsbeschreibungstabelle (Rückgabewerte beim OPEN WORKSTATION - Ruf)

Tabelle 5: Arbeitsstationsbeschreibungstabelle für Plotter K6411

Adresse	Inhalt	Bedeutung
intout(1)	24159	max. adressierbare Weite (x-Achse des Plotters in Rasterschritten
intout(2)	17599	max. adressiebare Höhe (y-Achse) des Plotters in Rasterschritten
intout(3)	0	Flag für Gerärekoordinatengenauigkeit
intout(4)	25	Weite eines Potterschrittes in Mikrometern
intout(5)	25	Höhe eines Plotterschrittes in Mikrometern
intout(6)	0	Anzahl von Zeichenhöhen (kontinuierlich einstellbar)
intout(7)	6	Anzahl von Linientypen
intout(8)	1	Anzahl von Linienstärken
intout(9)	5	Anzahl von Markentypen
intout(10)	0	Anzahl von Markenhöhen (kontinuierlich einstellbar)
intout(11)	2	Anzahl von Textarten (Fonts)
intout(12)	0	Anzahl von Mustern (Pattern)
intout(13)	0	Anzahl von Schraffuren (Hatch styles)
intout(14)	8	Anzahl der vordefinierten Farben
	a) b)	
intout(15)	5 0	Anzahl der GDP
intout(16)	1 -1	Rechteck GDP (BAR)
intout(17)	2 -1	Keisbogen-GDP (ARC)
intout(18)	3 -1	Kreissegment-GDP (PIE SLICE)
intout(19)	4 -1	Kreis-GDP (CIRCLE)
intout(20)	5 -1	Text-GDP (PRINT GRAPHIC CHARACTERS)
intout(21)..(25)	-1	(keine weiteren GDP definiert
intout(26)	3 -1	Attribut-Satz für Rechteck-GDP (entspr. fill area)
intout(27)	0 -1	Attribut-Satz für Kreisbogen-GDP (entspr. polyline)
intout(28)	3 -1	Attribut-Satz für Kreissegment-GDP (entspr. fill area)
intout(29)	3 -1	Attribut-Satz für Kreis-GDP (entspr. fill area)
intout(30)	2 -1	Attribut-Satz für Text-GDP (entspr. text)
intout(31)..(35)	-1	(keine weiteren GDP-Attribute)
intout(36)	1	Flag für Farbfähigkeit (ja)
intout(37)	1	Flag für Textdrehung (ja)
intout(38)	0	Flag für Flächendrehung (nein)
intout(39)	0	Flag für Pixeloperationen (nein)
intout(40)	8	Anzahl verfügbarer Farben
intout(41)	1	Anzahl der LOCATOR-Geräte

Tabelle 5: Fortsetzung

Adresse	Inhalt	Bedeutung
intout(42)	0	Anzahl der VALUATOR-Geräte
intout(43)	0	Anzahl der CHOICE-Geräte
intout(44)	0	Anzahl der STRING-Geräte
intout(45)	2	Arbeitsstationstyp (input/output)
ptsout(1)	0	
ptsout(2)	100	min. Zeichenhöhe in Geräteeinheiten (Plotterschritten)
ptsout(3)	0	
ptsout(4)	1000	max. Zeichenhöhe in Geräteeinheiten
ptsout(5)	1	min. Linienbreite in Geräteeinheiten
ptsout(6)	0	
ptsout(7)	1	max. Linienbreite in Geräteeinheiten
ptsout(8)	0	
ptsout(9)	0	
ptsout(10)	100	min. Markenhöhe in Geräteeinheiten
ptsout(11)	0	
ptsout(12)	1000	max. Markenhöhe in Geräteeinheiten

Die ptsout-Werte werden an den Nutzer transformiert in normalisierte Koordinaten (NDC) zurückgegeben.

Anmerkungen:
- Die Werte in intout(1), (2) gelten für den Fall, daß die gesamte Darstellungsfläche des Plotters K6411 genutzt wird. Wird dagegen mit einem Viewport, d.h. mit einem rechteckigen, achsenparallelen Teilbereich der Darstellungsfläche (siehe Punkt 3.6.5. in Anlage 4) gearbeitet, werden die entsprechenden Werte für das gewählte Viewport angegeben.
- Die o.g. Werte für minimale und maximale Zeichen- bzw. Markerhöhe sind zweckmäßige Empfehlungen aber keine einschränkenden Grenzen.
- Die Werte für intout(15)...(30) unter a) gelten für K6411SG.SYS, die unter b) für K6411S.SYS.

4.4. Attribute für die Darstellungselemente

a) Linienarten:
1 Vollinie
2 Strichlinie
3 Punktlinie
4 Strich-Punkt-Linie
5 Strichlinie (Langstrich-Kurzstrich)
6 Strichlinie (Langstrich-Kurzstrich-Kurzstrich)

b) Markentypen:
1 . (Punkt)
2 + (Plus-Zeichen)
3 * (Stern)
4 o (Kleinbuchstabe o)
5 x (Kleinbuchstabe x)

c) Schriftarten:
1 gerade Schrift (Neigung 0 Grad)

2 schräge Schrift (Neigung 18 Grad)

d) Farben: Die acht automatisch wechselbaren Schreibwerkzeuge können von unterschiedlicher Farbe oder unterschiedlicher konstruktiver Gestaltung sein (z.B. Faser-, Tinten-, Tuscheschreiber). Einer Werkzeug-Nr. kann eine Farbe zugeordnet werden, d.h. die Werkzeug-Nr. entspricht dem Farbindex.
Vorzugsbelegung: 1 rot
2 grün
3 blau
4 schwarz

4.5. ESCAPES

Von den in der SCP-GX-Beschreibung genannten ESCAPE-Funktionen wird nur die Funktion 1 (INQUIRE ADDRESSABLE CHARACTER CELLS) realisiert. Darüber hinaus werden zusätzliche ESCAPE-Funktionen definiert und im Driver realisiert, um spezielle Leistungen des Plotters K6411 auch über die SCP-GX-Schnittstelle verfügbar zu machen. Diese Funktionen werden im folgenden beschrieben.

4.5.1. SET CHARACTERSLANT

Neigungswinkel für Textzeichen einstellen.

Parameter und Funktionen siehe Punkt 3.5.1..

4.5.2. SET CHARACTER PROPORTION

Höhe und Breite für Textzeichen einstellen.

Parameter und Funktion siehe Punkt 3.5.2..

4.5.3. MOVE PLOT POSITION

Verändern der aktuellen Schreibposition im Zusammenhang mit Textausgabe.

Parameter und Funktion siehe Punkt 3.5.3..

4.5.4. SET CHARACTER SET

Einstellen Zeichensatz

Eingabe: contrl(1)=5
contrl(4)=1
contrl(6)=54
intin(1)=Zeichensatz-Nr.

Ausgabe: contrl(3)=0

Beschreibung: Entsprechend der angegebenen Zeichensatz-Nr. wird der Zeichensatz für folgende Textausgaben ausgewählt, der dann bis auf Widerruf gilt. Die Anzahl der verfügbaren Zeichensätze ist vom Ausstattungsgrad des Plotters abhängig.

Zeichensätze:		
	0,1,2,3,4	lateinische Zeichensätze
	10,11,12,13,14	> (Untersohiede bei Sonderzeichen)
	20	kyrillischer Zeichensatz

4.5.5. DRAW X-AXIS-TICK

Zeichnen eines Skalenstriches auf der x-Achse

Parameter und Funktion siehe Punkt 3.5.6..

4.5.6. DRAW Y-AXIS-TICK

Zeichnen eines Skalenstriches auf der y-Achse

Parameter und Funktion siehe Punkt 3.5.7..

4.5.7. PLOT RELATIV (POLYLINE)

Zeichnen von Polyline mit relativen Koordinaten.

Parameter und Funktionen siehe Punkt 3.5.10..

4.5.8. PLOT RELATIV (ARC)

Zeichnen eins Kreisbogens mit relativen Mittelpunktkoordinaten

Parameter und Funktion siehe Punkt 3.5.11..

4.5.9. SET WINDOW

Einstellen der Grenzen eines Fensters auf der Darstellungsfläche.

Parameter und Funktion siehe Punkt 3.5.12..

4.5.10. LOCATOR INPUT IN DC

Abfrage der Zeichenwerkzeugposition in Gerätekoordinaten

Eingabe: contrl(1)=5
contrl(6)=63
intin(1)=6 (Locator-Geräte-Nr. für Plotter)

Ausgabe: contrl(5)=2
contrl(3)=0
intout(1)=X-Koordinate des digitalisierten Punktes in DC
intout(2)=Y-Koordinate des digitalisierten Punktes in DC

Beschreibung: Bei Aufruf der Funktion wird der Plotter in den Digitalisierzustand versetzt (Anzeige "dP" an der Segmentanzeige). Über die Tasten zur Handsteuerung (Joystick) der Bewegung des Zeichenwerkzeuges wird die zu digitalisierende Position angefahren. Durch Betätigen der Taste "Enter" wird die aktuelle Zeichenwerkzeugposition aufgenommen und der String mit den digitalisierten Koordinatenwerten zur Übergabe an den Rechner bereitgestellt (Anzeige "dS" an der Segmentanzeige). Nach Übernahme dieses String durch den Rechner, geht der Plotter wieder in den on-line-Zustand zurück. Die in intout zurückgegebenen Koordinatenwerte sind DC-Werte und können z. B. als Viewport-Grenzen bei einem erneuten OPEN-Ruf verwendet werden.

4.5.11. UNIVERSAL ESCAPE-FUNCTION

Universelle ESCAPE-Funktion zur direkten Ausgabe von Plotterbefehlen.

Parameter und Funktion siehe Punkt 3.5.13..

4.6. Hinweise zur Arbeit mit SCP-GX-Funktionen für Ausgaben zum Plotter K6411

4.6.1. Koordinatenkorrektur

Der Plotter K6411 hat in x-Richtung 24160 und in y-Richtung 17600 adressierbare Punkte.
Das bedeutet, daß eine in NDC-Einheiten gegebene Strecke auf dem Plotter in x-Richtung größer abgebildet wird als in y-Richtung. Die Ursache dafür ist die SCP-GX-Konvention, daß die NDC-Koordinaten 0...32767 vollständig auf jede Achse des Ausgabegerätes abgebildet werden. Soll also auf einem Ausgabegerät mit unterschiedlicher maximaler Anzahl von Rasterschritten in x- und y-Richtung eine unverzerrte, d.h. in x- und y-Richtung äquidistante Darstellung erfolgen, muß im Anwenderprogramm eine Koordinatenkorrektur für eine Achse vorgesehen werden.

ykor = 24160/17600 = 1.373

und analog bei Korrektur der x-Koordinaten

xkor = 17600/24160 = 0.728

mit den entsprechenden Koordinatenwerten zu multiplizieren.

4.6.2. Textdarstellung

Textzeichen werden von der Plotter-Firmware erzeugt, wobei die Zeichensätze mit festen Zeichenboxen (Computerschrift, Zeichensatz 0...4) ein Raster von 6 x 16 für die Zeichnbox und 4 x 8 für das Zeichen verwenden. Der Abstand zwischen 2 Zeichen beträgt eine halbe Zeichenbreite, der Zeilenabstand eine ganze Zeichenhöhe.
Mit der SCP-GX-Funktion 12 (SET CHARACTER HEIGHT) kann die Zeichenhöhe bei konstanter Proportion und mit der Funktion 5 (ESCAPE), Subfunktion 52 (SET CHARACTER PROPORTION) können Größe und Proportion der Zeichen verändert werden.
Die Standardgröße der Zeichen beträgt bei Nutzung der gesamten Plotfläche 0,285 cm in der Breite (einschließlich Lücke zum

nächsten Zeichen) und 0,375 cm in der Höhe.
Mit der SCP-GX-Funktion 13 (SET CHARACTER UP VECTOR) wird die Richtung der Basislinie für nachfolgende Textausgaben festgelegt. Dabei wird in intin(1) der Winkel zwischen Basislinie und positiver x-Achse (gemessen entgegen dem Uhrzeigersinn) für die Ermittlung der Schreibrichtung verwendet, wenn die Parameter in intin(2) und intin(3) bei Funktionsaufruf den Wert 0 haben. Im anderen Fall wird die Schreibrichtung aus intin(2) und intin(3) ermittelt.
Bei Ausgabe von Text über SCP-GX-Funktion 8 bzw. 11 wird für diese Ausgabe in den kyrillischen Zeichensatz umgeschaltet, wenn mindestens 1 Zeichen im Ausgabestring ein gesetztes 8. Bit (alternativer Zeichensatz) hat. Eine gemischte Ausgabe von lateinischen und kyrillischen Zeichen innerhalb desselben String ist deshalb nicht möglich.

4.6.3. GDP für Kreisbogen bzw. Kreissegment

Varianten der Parameterübergabe siehe Punkt 3.6.3..

4.6.4. Plotten mit relativen Koordinaten

Abweichend von den SCP-GX-Konventionen werden beim Plotten mit relativen Koordinaten (siehe Punkt 3.5.10. und 3.5.11.) diese Koordinaten als vorzeichenbehaftete NDC-Werte in intin-Bereich übergeben, da im ptsin-Bereich nur positive Werte (0...32767) zugelassen sind. Die Transformation NDC -> DC wird in diesem Fall im Plotter-Driver durchgeführt.

4.6.5. Nutzung eines Viewports

Einstellung und Wirkungsweise siehe Punkt 3.6.5.

4.7. Zusammenstellung der verwendbaren SCP-GX-Funktionen

```
 1 OPEN WORKSTATION
 2 CLOSE WORKSTATION
 3 CLEAR WORKSTATION
 4 UPDATE WORKSTATION
 5 ESCAPE (mit den Funktionen 1, 51...54, 56, 57, 60...63, 99)
 6 POLYLINE
 7 POLYMARKER
 8 TEXT
 9 FILLED AREA          | nur
10 CELL ARAY            | Umrandungen
11 GDP (Rechteck, Kreisbogen, Kreissegment, Kreis, Text)
12 SET CHARACTER HEIGHT
13 SET CHARACTER UP VECTOR
14 SET COLOR REPRESENTATION
15 SET POLYLINE LINETYPE
17 SET POLYLINE COLOR INDEX
18 SET POLYMARKER TYP
19 SET POLYMARKER SCALE
20 SET POLYMARKER COLOR INDEX
21 SET TEXT FONT
22 SET TEXT COLOR INDEX
23 SET FILL INTERIOR STYLE
```

25 SET FILL COLOR INDEX
26 INQUIRE COLOR REPRESENTATION
28 INPUT LOCATOR
33 SET INPUT MODE

Anlage 5 Drivertestprogramm DRVTST.CMD

1. Arbeitsweise und Aufruf von DRVTST

Das Programm DRVTST.CMD dient zum Testen von Gerätedrivern für die Grafikerweiterung SCP-GX des Betriebssystems SCP 1700.

Voraussetzungen für den Start von DRVTST.CMD sind:
1. Laden der SCP-GX-Datei GRAPHICS.CMD
   Kommando: GRAPHICS [d:]
   d - optionales Diskettenlaufwerk, von dem die Datei ASSIGN.SYS und die Gerätedriver dateiname.SYS geladen werden sollen. Fehlt diese Angabe, wird auf das Standardlaufwerk zugegriffen.
2. Vorhandensein der Datei ASSIGN.SYS auf einer Platte im unter "d:" angesprochenen Laufwerk.
3. Vorhandensein der Dateien dateiname.SYS auf der gleichen Platte. Außerdem müssen diese Dateinamen in ASSIGN.SYS enthalten sein.

Das Gerätedriver-Testprogramm wird mit dem Kommando
DRVTST
gestartet. Das Programm meldet sich mit der Ausschrift
TEST PROGRAM FOR SCP-GX-FUNCTIONS

Daraufhin und nach jeder getesteten SCP-GX-Funktion erscheint die Eingabeanforderung
SCP-GX-function or help (<CR>) or parameter values (V):

Die Eingaben erfolgen entsprechend der "Programmtechnischen Beschreibung SCP-GX". Für jede SCP-GX-Funktion werden bestimmte Elemente der Felder contrl
intin
ptsin
mit Werten belegt.
Nach Ausführung der SCP-GX-Funktion werden vom jeweiligen Gerätedriver bestimmte Elemente der Felder contrl
intout
ptsout
gesetzt.

In den Abschnitten 2 und 3 werden zwei verschiedene Möglichkeiten beschrieben, die Eingabewerte der SCP-GX-Funktionen zu setzen.

Die maximale Anzahl der benutzbaren contrl-Parameter beträgt 10; die der intin- und intout-Parameter 255 und die der ptsin- und ptsout-Parameter 120. Der Wertebereich für Eingabewerte liegt zwischen -32767 und +32767. Werden außerhalb dieses Wertebereichs liegende Zahlen eingegeben, so werden diese Eingaben von DRVTST entweder auf -32767 oder auf +32767 gesetzt. Für Punktkoordinaten werden jedoch vom SCP-GX nur Werte im Bereich [0,32767] verarbeitet.

2. Testen von SCP-GX-Funktionen

Wird die Eingabeanforderung
SCP-GX-function or Help (<CR>) or parameter values (V):

mit Eingabe einer Zahl zwischen 1 und 33 beantwortet, so wird die entsprechende SCP-GX-Funktion ausgeführt, die in der "Programmtechnischen Beschreibung SCP-GX" beschrieben wird. Die weiteren Parameter werden daran anschließend entsprechend der ausgewählten SCP-GX-Funktion eingegeben. Dabei ist es äquivalent, ob ein <CR> oder ein Komma als Trennzeichen angegeben wird. Wird die Eingabe eines Parameters mit <CR> abgeschlossen, erscheint auf dem Bildschirm die Ausschrift, welcher Parameter als nächstes einzugeben ist. Dadurch ist es möglich alle Parameter einer SCP-GX-Funktion im Dialog einzugeben.

Durch Eingabe eines Kommas als Trennzeichen zwischen den Eingabeparametern ist es möglich alle Eingabeparameter einer SCP-GX-Funktion auf einer Kommandozeile anzugeben.
Zum Beisiel ist äquivalent:

```
SCP-GX-function or help (<CR>) or parameter values (V): 6
                                                         -

Anzahl der Punkte: 2
                   -

Points: x1,y1,...,xn,yn:
ptsin(1) = 10000
           -----
ptsin(2) = 10000
           -----
ptsin(3) = 20000
           -----
ptsin(4) = 20000
           -----
```

und

```
SCP-GX-function or help (<CR>) or parameter values (V): 6,2,
10000,10000,20000,20000                                  ----
-----------------------
```

Wird als Trennzeichen das Komma verwendet und werden zuwenig Parameter eingegeben, werden diese den entsprechenden Feldelementen zugeordnet und die Eingabeanforderung des nächsten Parameters erscheint auf dem Bildschirm. Werden zuviel Parameter auf einer Kommandozeile eingegeben, werden die restlichen ignoriert.

Text- und andere Zeicheneingaben bei den SCP-GX-Funktionen: Text, GDP-Funktion 5 und ESCAPE-Funktion 12 können nicht wie Zahlenwerte durch Komma getrennt eingegeben werden, sondern nur auf Eingabeanforderung.

Ebenso kann bei "OPEN WORKSTATION" die Anfrage, ob die Werte für intin(2) bis intin(10) explizit gesetzt werden sollen, nur nach Anforderung beantwortet werden. Falls diese Anforderung negativ beantwortet wird, werden diese Werte mit "1" initialisiert.

Wird eine der ESCAPE-Funktionen 4-12 eingegeben, lassen sich andere ESCAPE-Funktionen direkt eingeben, ohne das vorher die "5" für ESCAPE angegeben wird. Damit soll das Ergebnis dieser ESCAPE-Funktionen beim Test von Bildschirmdrivern besser sichtbar gemacht werden.

Weiterhin ist es möglich, ESCAPE-Funktionen, die nicht in der "Programmtechnischen Beschreibung SCP-GX" aufgeführt sind, zu testen. Das Programm meldet dann die in contrl(6) eingetragene ESCAPE-Funktionsnummer zurück. Daraufhin wird der Wert von

contrl(4) eingegeben. Die folgenden intin-Parameter sind, durch Komma getrennt, in einer Kommandozeile einzugeben (maximal 255 Zeichen), da die Anzahl der intin-Werte für jede zusätzliche ESCAPE-Funktion unterschiedlich sein kann. Die Punktkoordinaten können dann entweder im Dialog oder auf einer Kommandozeile eingegeben werden.

Durch einige SCP-GX-Funktionen wird, falls ein Bildschirmdriver getestet werden soll, auf den Grafikbildschirm gezeichnet. Das Umschalten von Alphanumerik- auf Grafikbildschirm wird selbständig von DRVTST.CMD realisiert bei Arbeitsstationsnummer 1-10 (Bildschirmdriver). Das Zurückschalten auf den Alphanumerikbildschirm geschieht durch Eingabe von <CR>.
Zu jeder SCP-GX-Funktion erscheinen die Rückkehrwerte entsprechend der "Programmtechnischen Beschreibung auf dem Bildschirm.
Vor Aufruf einer SCP-GX-Funktion wird der Wert von contrl(3) auf "-1" gesetzt. Da alle SCP-GX-Funktionen contrl(3) als Rückkehrwert benutzen, besteht dadurch eine effektive Kontrolle, ob die jeweilige SCP-GX-Funktion auch ausgeführt wurde. Nach Ausführung einer SCP-GX-Funktion und Ausgabe der Rückkehrwerte auf dem Bildschirm werden contrl[5], sowie alle intout- und ptsout-Parameter mit 0 initialisiert.

Wird bei Anforderung
  SCP-GX-function or Help (<CR>) or parameter values (V):
keine Zahl zwischen 1 und 33 eingegeben und werden auch die in Abschnitt 3 erläuterten Zeichen nicht eingegeben, so erfolgt vom Programm die Fehlermeldung:
  Unallowed SCP-GX-function
und daraufhin die Eingabeanforderung für die nächste SCP-GX-Funktion.

Als erste SCP-GX-Funktion sollte grundsätzlich nach Start des Programmes "OPEN WORKSTATION" durchgeführt werden.

Wird nach Eingabe eines Wertes zur Auswahl der SCP-GX-Funktion die Eingabe eines nachfolgenden Eingabeparameters mit <CR> oder Komma beantwortet, so bleibt der jeweilige Eingabeparameter unverändert. Auf diese Art und Weise braucht z.B. eine größere Anzahl von Punktkoordinaten nicht mehrmals eingegeben werden.

3. Testen im V-Modus

Die zweite Möglichkeit besteht darin, nach der Anforderung
  SCP-GX-function or help (<CR>) or parameter values (V):
ein "V" einzugeben. Dies ermöglicht die Auflistung einzelner Parameterfelder auf dem Bildschirm oder die Änderung einzelner Eingabefeldelemente. Nach Eingabe von "V" erscheint ein Menü auf dem Bildschirm zur Eingabe der gewünschten Funktion:

```
 1  print contrl
 2  print intin
 3  print ptsin
 4  print intout
 5  print ptsout
 6  all values = 0
 7  set contrl
 8  set intin
 9  set ptsin
10  graphic call
11  graphic call with graphic mode
```

Bei Wahl einer "Print"-Funktion werden die jeweiligen Parameterfelder vom Index 1 bis zum einzugebenden Index aufgelistet, während eine "Set"-Funktion nur den jeweiligen Wert des durch Index angesprochenen Feldelementes ändert. Außerdem können sämtliche Werte der Parameterfelder auf Null gesetzt werden. Nachdem alle Eingabeparameter gesetzt wurden, kann der GDOS-Aufruf entweder mit oder ohne Umschalten auf den Grafikbildschirm ausgeführt werden. Der V-Modus ermöglicht es spezielle Driverfunktionen zu testen oder sich jederzeit einen Überblick über die Parameterwerte zu verschaffen. Hinter "V" können auch alle weiteren Eingaben durch Komma getrennt auf einer Kommandozeile erscheinen. Erfolgt eine Eingabe, die nicht im Menü vorhanden ist, wird der V-Modus verlassen.
Der Wertebereich für jedes durch Funktion 7,8 bzw. 9 zu initialisierende Feldelement von contrl, intin bzw. ptsin beträgt [-32767,32767].
Ein Beispiel für eine Eingabe im V-Modus folgt:

```
  SCP-GX-function or help (<CR>) or function values (V): V,7,1,1,
                                                         --------
  7,2,0,7,4,10,8,1,1,10,7,1,5,7,4,0,7,6,3,8,1,0,10,4,45,5,12
  ----------------------------------------------------------
```

Dabei werden folgende Funktionen ausgeführt:

```
7,1,1 : contrl(1)=1
7,2,0 : contrl(2)=0
7,4,10: contrl(4)=10
8,1,1 : intin(1)=1
   10 : __BDOS(0x473,par)  (par:Startadresse des Parameterblocks)
7,1,5 : contrl(1)=5        (0x473 entspricht 473H)
7,4,0 : contrl(4)=0
7,6,3 : contrl(6)=3
8,1,0 : intin(1)=0
   10 : __BDOS(0x473,par)
 4,45 : Auslisten von intout(1),...,intout(45) auf dem Bildschirm
 5,12 : Auslisten von ptsout(1),...,ptsout(12) auf dem Bildschirm
```

Das heißt, es wird "OPEN WORKSTATION" mit einem Bildschirmdriver ausgeführt. Im Ergebnis dieser Funktion wird auf dem Grafikbildschirm umgeschaltet. Durch die ESCAPE-Funktion "EXIT GRAPHICS MODE" wird auf den Alphanumerikbildschirm zurückgeschaltet. Anschließend werden intout- und ptsout-Werte ausgegeben.

4. Help-Funktion

Wird die Anforderung

```
  SCP-GX-function or help (<CR>) or parameter values (V):
```

mit <CR> beantwortet, zeigt das Programm das Help-Menü für die Auswahl von SCP-GX-Funktionen entsprechend der "Programmtechnischen Beschreibung SCP-GX" an, daraufhin können die entsprechenden Untermenüs durch Eingabe eines Buchstabens ohne <CR> auf dem Bildschirm aufgelistet werden. Durch <CR> wird die Help-Funktion wieder verlassen.

Anlage 6 Nutzung von SCP-GX unter FORTRAN-77

Der Liefersatz für FORTRAN-77 (speziell die Diskette mit den Laufzeitbibliotheken) enthält Interfaceprogramme für den Aufruf von SCP 1700 - und SCP-GX-Systemleistungen durch FOR77-Programme. Diese Programme werden durch gleichfalls auf dieser Diskette vorhandene Textdateien beschrieben. Die Namen der Interfaceprogramme und der zugehörigen Textdateien sind in der Datei F77INFO.TXT verzeichnet. Der nachfolgende Text entspricht dem Inhalt der Textdatei GDOS86.TXT.

Das Unterprogramm GDOS86 ermöglicht den Aufruf des Grafik-Systems SCP-GX von Programmen aus, die mit Hilfe des FORTRAN-77-Compilers FOR77 erzeugt wurden.

Dem Unterprogramm GDOS86 werden 5 Felder uebergeben. Der Datentyp dieser Felder ist INTEGER*2.

Aufruf des Unterprogrammes:

```
   CALL GDOS86(CONTRL,INTIN,PTSIN,INTOUT,PTSOUT)
```

Das Feld CONTRL enthält im ersten Element den Operationscode für die auszuführende Funktion des Grafik-Systems. Die Feldgrenzen sind abhängig von der auszuführenden Funktion. In der Anwendungstechnischen Beschreibung des SCP-GX ist der Inhalt der Felder beschrieben.

Beispiel: Eröffnen einer Arbeitsstation (Workstation)

```
          .
          .
          .
   INTEGER*2 CONTRL(5),INTIN(10),PTSIN(1),INTOUT(45),PTSOUT(12)
          .
          .
          .
C     Eingabeparameter  OPEN Workstation
   DATA CONTRL /1,0,6,10,48/
C     Workstation ist Bildschirm, durchgängiger Linienzug,
C     + als Marker, ...
   DATA INTIN /1,1,1,2,1,1,1,1,1,1/
C
C     Aufruf der OPEN-Funktion des SCP-GX
C
   CALL GDOS86(CONTRL,INTIN,PTSIN,INTOUT,PTSOUT)
          .
          .
          .
```

Die Felder INTOUT und PTSOUT enthalten nach dem OPEN-WORKSTATION die Spezifikationen der Arbeitsstation.

Auf Grund der Parametervermittlung des Compilers FOR77 kann eine SUBROUTINE auch als FUNCTION aufgerufen werden (allerdings nicht gemischt in einer FORTRAN=Programmeinheit). Auf diese Weise kann Programmcode zur Parametervermittlung eingespart werden, wenn der Aufruf als Anweisungsfunktion erfolgt. Der Funktionswert ist

undefiniert. Falls keine Typvereinbarung der Routine GDOS86 und der Anweisungsfunktion als INTEGER existiert, kann es zu Konvertierungsfehlern führen.

Beim Linken des Anwendungsprogrammes ist der Objektmodul GDOS86.OBJ einzufügen.

Vor dem Aufruf des Anwendungsprogrammes müssen das SCP-GX und ggf. die notwendige Firmware geladen werden.

Zur Unterstützung der Programmierung existieren weitere Unterprogramme:

- Eröffnen einer Arbeitsstation WSOPEN
- Schliessen/Löschen der Arbeitsstation WSCLCS
- ESCAPE-Funktión WSESCP
- Textausgabe WSCHAR

WSOPEN

Aufruf: CALL WSOPEN(WSN,INTOUT,PTSOUT)

Parameter:

```
INTEGER*2  WSN,INTOUT(45),PTSOUT(12)

WSN  ...  Arbeitsstationsidentifikation
          z.B.  1 Bildschirm
               11 Plotter
               21 Drucker
INTOUT,PTSOUT Rückgabeparameter
```

WSCLCS

Aufruf: CALL WSCLCS(n)

```
n = 2   CLOSE Workstation
n = 3   CLEAR Workstation
```

WSESCP

Aufruf: CALL WSESCP(n)

```
n = 2   Alpha Mode -> Graphic Mode des Bildschirmes
n = 3   Graphic Mode -> Alpha Mode des Bildschirmes
```

Weitere ESCAPE-Funktionen, die keine Parameter benoetigen, sind aufrufbar.

WSCHAR

Aufruf: CALL WSCHAR(L,X,Y,STRING)

Parameter:

```
INTEGER*2 L,X,Y
CHARACTER*( n ) STRING    mit    1<= n <= 80
```

L Länge des auszugebenden Textes (1 bis 80)

X,Y XY-Koordinaten des linken unteren Zeichenfeldes

STRING auszugebender Text

Abkürzungsverzeichnis

BIOS	Basic-I/O-System
BP	Basisanzeiger
DC	Gerätekoordinaten
DS	Datensegment
GBS	Grafisches Bildschirmsystem
GDOS	Graphics Device Operating System
GDP	Generalized Drawing Primitive
GIOS	Graphics Input/Output System
ID	Identifikationsnummer
NDC	Normalisierte Koordinaten
SCP-GX	graphische Systemerweiterung des SCP 1700
VGI	Virtuelles Geräteinterface

Sachwortverzeichnis

Arbeitsstation 7, 70
Arbeitsstation-ID 13, 14, 22, 68, 70

Basisanzeiger 8

Datensegment 8

Farbindex 59
Füllart 58
Funktionscode 7, 22, 68

GDOS-Ruffolge 36
Gerätekoordinaten 7, 68

Identifikationsnummer 7

Normalisierte Koordinaten 7, 9, 36, 68
Normalisierter Koordinatenraum 69

Operationscode 68, 70

Polygon 50

Stackbereich 22
Standardgerätedriver 21, 22

Virtuelle Geräteinterface 7, 16, 17, 70

Zeichenmatrix 54
Zellmatrix 51, 61
Zuweisungstabelle 13, 19, 20, 27, 68